Standard Archive Format for Europe

ERS Specialisation for Level 0 products

Reference	PDGS-SAFE-GMV-ERS-L0	issue 1	revision 1
Author(s)	GMV	Date: 24/06/2015	
Reviewed by	European Space Agency (ESA)	date	
Approved by		date	

ESRIN
Via Galileo Galilei - Casella Postale 64 - 00044 Frascati - Italy
Tel. (39) 06 94J80J - Fax (39) 06 94J80 280

Table of contents

List of Tables

List of Figures

1. Introduction

1.1. Purpose and scope

This document is part of the Standard Archive Format for Europe specialisation for ERS (SAFE Specialisation for ERS). This specialisation consists of the following set of documents:

- the ERS mission specialisation control book, which is the top-level document of the specialisation, containing all the information that is common to all SAFE ERS products and auxiliary files.
- one ERS product specialisation control book for ERS Level-0 products and auxiliary files.

The current book is the specialisation control book for ERS Level-0 products and auxiliary files.

1.2. Book organisation

The specialisation control book for ERS Level-0 products and auxiliary files is organized as follows:

Chapter 1: Introduction	Introductory part of the document.
Chapter 2: Target of preservation	Description of the target of preservation for L0 products and auxiliary files.
Chapter 3: Data Structures	Specification of the simple and complex types that are used to represent the structure of the products and auxiliary file types independently of the ERS instrument to which they are associated.

1.3. Acronyms and abbreviations

DFDL	Data Format Description Language
DLT	Digital Linear Tape
ENVISAT	ENVIronmental SATellite
GNU	GNU is Not Unix
MPH	Main Product Header
PDS	Payload Data Segment
SPH	Specific Product Header
W3C	World Wide Web Consortium
XML	eXtensible Mark-up Language

2. Target of preservation

Some of the ERS L0 products and auxiliary files in native format are available in tar/gzip format (with ".TGZ" extension i.e. GNU-zipped tar file merging) or alternatively in the UNIX compressed format (".Z" extension). However, the targets of preservation considered in this SAFE specialisation are the contents of those compressed files, i.e. the files which are stored within the tar/gzip or Z files.

Any product in native format must be unpackaged and decompressed before being converted into SAFE and the SAFE Packages will only contain the unpackaged and decompressed files. This is because the representation information schemas that are provided along with this specialisation describe the unpackaged and decompressed files, not the tar/gzip format (there would be limitations in doing this, as explained in the SAFE Core Specifications).

The following table summarises the list of product types in scope and points the reader to the sections in the document where the information about the main structure of the file can be found:

File Types	Structure specification
SAR_IM__0P	See section 3.2.1
SAR_EWA_0P	See section 3.2.2
PREC	See section 3.3.1.1
ORPM	See section 3.3.1.2
ORRM	See section 3.3.1.3
PATC	See section 3.3.2.1

Table 1: File Types Specification Index

3. Data Structures

The information included in this chapter has been generated using the specifications defined by the DFDL schemas that represent the structure of the L0 products and auxiliary files.

The representation information is described by means of complex structures that make use of simple types to represent the whole content of a given file type. The following sub-sections provide a detailed description of those complex/simple types.

The diagrams included in this document provide an overview of the structure of the products by depicting the schemas which provide their representation information.

3.1. Common Data Structures

3.1.1. Simple Types

The DFDL schemas used to represent the structure of the L0 products and auxiliary files make use of the standard W3C simple types (e.g. xs:string, xs:integer, xs:NCName, etc…).

Some of these types have been restricted for ERS needs, resulting in new specific types detailed below.

3.1.1.1. uc

Base Type	Length (bytes)	Comments
xs:string	1	Unsigned char (uc)

Table 2: uc Specification

3.1.1.2. utc

Base Type	Length (bytes)	Comments
xs:dateTime	27	UTC time (27 bytes) dd-MMM-yyyy hh:mm:ss.uuuuuu Pattern: dd-MMM-yyyy HH:mm:ss.SSSSSS

Table 3: utc Specification

3.1.1.3. int_s04d

Base Type	Length (bytes)	Comments
xs:short	4	Pattern: +000;-000

Table 4: int_s04d Specification

3.1.1.4. int_s06d

Base Type	Length (bytes)	Comments
xs:int	6	Pattern: +00000;-00000

Table 5: int_s06d Specification

3.1.1.5. int_06d

Base Type	Length (bytes)	Comments
xs:int	6	Pattern: 000000

Table 6: int_06d Specification

3.1.1.6. int_s07d

Base Type	Length (bytes)	Comments
xs:int	7	Pattern: +000000;-000000

Table 7: int_s07d Specification

3.1.1.7. int_07d

Base Type	Length (bytes)	Comments
xs:int	7	Pattern: 0000000

Table 8: int_07d Specification

3.1.1.8. int_s11d

Base Type	Length (bytes)	Comments
xs:long	11	Pattern: +0000000000;-0000000000

Table 9: int_s11d Specification

3.1.1.9. int_s21d

Base Type	Length (bytes)	Comments
xs:unsignedLong	21	Pattern: +00000000000000000000; -00000000000000000000

Table 10: int_s21d Specification

3.1.1.10. real_s08_6f

Base Type	Length (bytes)	Comments
xs:decimal	8	Pattern: +.00000;-.00000

Table 11: real_s08_6f Specification

3.1.1.11. real_s11_6f

Base Type	Length (bytes)	Comments
xs:decimal	11	Pattern: +0000.000000;-0000.000000

Table 12: real_s11_6f Specification

3.1.1.12. real_11_6f

Base Type	Length (bytes)	Comments
xs:decimal	11	Pattern : 0000.000000

Table 13: real_11_6f Specification

3.1.1.13. real_s12_3f

Base Type	Length (bytes)	Comments
xs:decimal	12	Pattern : +0000000.000;-0000000.000

Table 14: real_s12_3f Specification

3.1.1.14. real_s12_6f

Base Type	Length (bytes)	Comments
xs:decimal	12	Pattern : +0000.000000;-0000.000000

Table 15: real_s12_6f Specification

3.2. ERS L0 Products

The following subsections provide a description of the data structures for the ERS SAR Image (SAR_IM__0P) and ERS SAR Wave (SAR_EWA_0P) products in scope.

3.2.1. SAR_IM__0P

The ERS SAR Image product in WILMA format contains a complete transcribed satellite imaging sequence (a passage), with all related information. This information is split in the following files:

- Pass Identification Header (DTPassID)
- Sensor Acquired Data (DTVideoData)
- User Header (DTUserHeader)
- Segment Descriptor (DTSegment)
- Orbit Data (DTOrbitFile) - optional
- Statistics file (DTStatisticFile).

3.2.1.1. DTPassID

The DTPassID (Pass Identification Header) contains the information to unambiguously identify the imaging sequence contained in the product.

This record is divided into five logical sections:

- Mission and Instrument Identification
- Ground Stations and Transcription System Identification
- Transcription Identification
- Orbit and Acquisition Identification
- Pointers to Tape Data Structure

The next figure provides a high level overview of the complex structures used to represent the information of the DTPassID file:

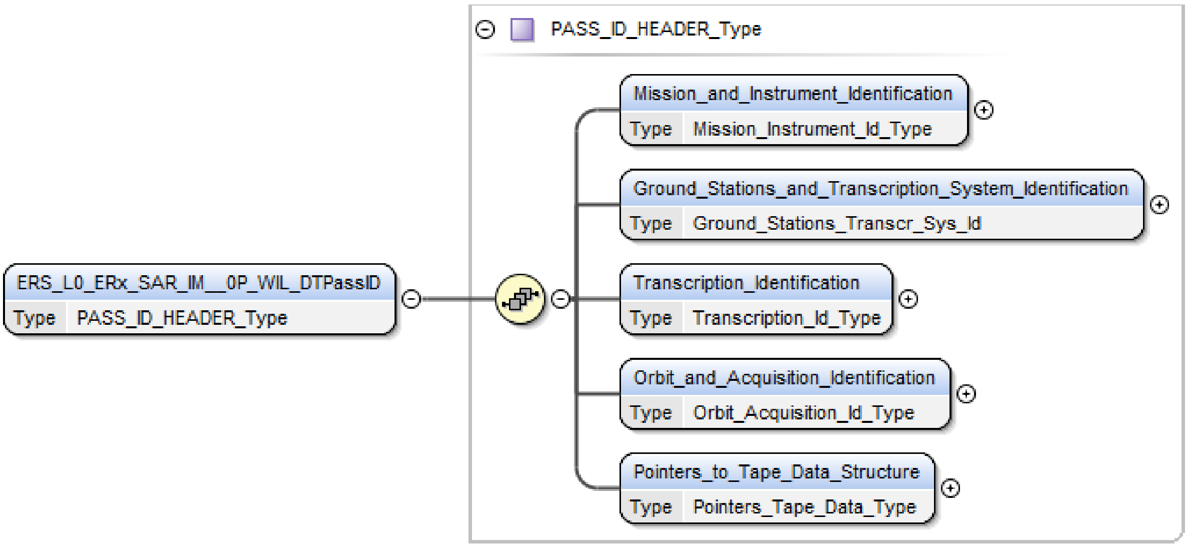

PASS_ID_HEADER_Type
ERS_L0_ERx_SAR_IM__0P_WIL_DTPassID
Type PASS_ID_HEADER_Type
Mission_and_Instrument_Identification
Type Mission_Instrument_Id_Type
Ground_Stations_and_Transcription_System_Identification
Type Ground_Stations_Transcr_Sys_Id
Transcription_Identification
Type Transcription_Id_Type
Orbit_and_Acquisition_Identification
Type Orbit_Acquisition_Id_Type
Pointers_to_Tape_Data_Structure
Type Pointers_Tape_Data_Type

Figure 1: DFDL schema organisation for SAR_IM__0P product in WILMA format - DTPassID

3.2.1.1.1. Root Element

#	Name/Description	Format
1	**ERS_L0_ERx_SAR_IM__0P_WIL_DTPassID** ERS SAR IM Level-0 Product File in WILMA Format PASS IDENTIFICATION HEADER file The "Pass Identification Header" is the first file of the Transcription Area. It contains the information, available at the beginning of the transcription, to unambiguously identify the imaging sequence contained in the file. This record is divided into five logical sections: 1. Mission and Instrument Identification 2. Ground Stations and Transcription System Identification 3. Transcription Identification 4. Orbit and Acquisition Identification 5. Pointers to Tape Data Structure	PASS_ID_HEADER_Type

Table 16: ERS_L0_ERx_SAR_IM__0P_WIL_DTPassID Specification

3.2.1.1.2. Complex Types

3.2.1.1.2.1.PASS_ID_HEADER_Type

#	Name/Description	Format
1	**Mission_and_Instrument_Identification**	Mission_Instrument_Id_Type
2	**Ground_Stations_and_Transcription_System_Identification**	Ground_Stations_Transcr_Sys_Id
3	**Transcription_Identification**	Transcription_Id_Type
4	**Orbit_and_Acquisition_Identification**	Orbit_Acquisition_Id_Type

#	Name/Description	Format
5	**Pointers_to_Tape_Data_Structure**	Pointers_Tape_Data_Type

Table 17: PASS_ID_HEADER_Type Specification

3.2.1.1.2.2.Mission_Instrument_Id_Type

Mission and Instrument Identification

#	Name/Description	Format
1	**Reserved_1** Reserved	xs:hexBinary 76bytes
2	**Satellite_ID** Satellite Code: 1 - LANDSAT 2 - MOS 3 - J-ERS 4 - SPOT 5 - ERS 6 - IRS-C 7 - RADARSAT 8 - NOAA 9 - RESERVED 10 - RESERVED 11 - HELIOS 12 - SHUTTLE 13 - EOSAM 14 - EOSPM	xs:unsignedShort 2 bytes
3	**Mission_ID** Satellite Mission Number	xs:unsignedShort 2 bytes
4	**Instr_Type_ID** Satellite Mission Number: 1 - LANDSAT MSS 2 - LANDSAT TM 3 - LANDSAT ETM 4 - LANDSAT RBV 5 - MOS MESSR 6 - J-ERS VNIR 7 - J-ERS SWIR 8 - Not Used 9 - Not Used 10 - ERS AMI SAR 11 - ERS ATSR 12 - SPOT HRV 13 - J-ERS SAR 14 - NOAA AVHRR 15 - SPOT HRVIR 16 - SHUTTLE XSAR 17 - MODIS	xs:unsignedShort 2 bytes
5	**Reserved_2** Reserved	xs:unsignedShort Min Occurs: 2

#	Name/Description	Format
		Max Occurs: 2 2 bytes

Table 18: Mission_Instrument_Id_Type Specification

3.2.1.1.2.3.Ground_Stations_Transcr_Sys_Id

Ground Stations and Transcription System Identification

#	Name/Description	Format
1	**Station_ID** Acquisition Ground Station Code: 67 - Adelaide 97 - Agrhymet 10 - AliceSpring 68 - Aspendale 105 - Atlanta 20 - Aussaguel 6 - Bangkok 52 - BantonRouge 49 - Bedford 74 - Beijing 30 - Berlin 31 - Berne 112 - Bishkek 29 - Bremenhaven 28 - Budapest 96 - Cairo 92 - Casey 64 - Cashoiera 32 - Copenhagen 103 - Cordoba 8 - Cotopaxi 23 - Cuiaba 76 - Da-Xing 33 - DeBilt 75 - Dhaka 50 - Downsview 34 - Dundee 51 - Edmonton 7 - Fairbanks 24 - Farnborough 35 - Frascati 1 - Fucino 65 - Funceme 9 - Gatineau 54 - GilmoreCreek 36 - Hamburg 19 - HarteBeesHoek 16 - Hatoyama 37 - Helsinki	xs:short 2 bytes

#	Name/Description	Format
	22 - Hobart 69 - Honolulu 11 - Hyderabad 21 - Islamabad 115 - Itu 77 - Keelung 116 - KhantyMansiysk 2 - Kiruna 78 - Kiyose 113 - Kitab 111 - Kourou 38 - Krakow 91 - KualaLumpur 12 - Kumamoto 39 - Lannion 79 - Lapan 99 - LaReunion 40 - Lasham 106 - Libreville 41 - Madrid 109 - Malindi 80 - Manila 3 - Maspalomas 104 - Matera 93 - McMurdo 53 - Miami 110 - Moscow 98 - Nairobi 55 - NESDIS 108 - Neustrelitz 81 - NewDelhi 102 - Norman 42 - Norrkoping 27 - Oberpfaff 43 - Offenbach 15 - OHiggins 44 - Oslo 94 - Palmer 18 - PariPari 70 - Perth 45 - Prague 25 - Pretoria 13 - PrinceAlbert 57 - RedwoodCity 82 - Riyadh 46 - Rome 101 - RRSC_Nairobi 66 - Santiago 26 - Scanzano 58 - Scipps Inst	

#	Name/Description	Format
	84 - Selangor 87 - Sendai 85 - Seoul 86 - SeoulUniv 83 - Singapore 59 - SiouxFalls 47 - Spitzenbergen 60 - StennisSpace 71 - Sydney 17 - Syowa 90 - Taipei 5 - Tel_Aviv 95 - TerranovaBay 88 - TokaiUniv 89 - TokyoUniv 72 - Townsville 48 - Traben-Trar 4 - Tromso 100 - Tunis 114 - UlanBator 61 - UnivOfAlaska 63 - UnivOfRhodeIsl 62 - UnivOfTexas 56 - WallopsIsl 73 - Wellington 14 - WestFreugh	
2	**Station_DT_ID** Data Transcription Ground Station Code	xs:short 2 bytes
3	**Filler** Padding for stucture alignment	xs:hexBinary 2bytes

Table 19: Ground_Stations_Transcr_Sys_Id Specification

3.2.1.1.2.4.Transcription_Id_Type

Transcription Identification

#	Name/Description	Format
1	**Source_Type** Data Source Type: 1 - AMPEX 14 tracks 2 - Shlumberger 14 tracks 3 - Shlumberger 42 tracks 4 - Penny and Giles 5 - Honeywell HD-96 6 - AMPEX DCRSi 7 - CREO Optical Tape 8 - Direct Ingestion 9 - SONY DIR 1000 (R)	xs:unsignedInt 4 bytes
2	**Format_SyncType** Format Synchroniser/Decommutator Code:	xs:unsignedInt 4 bytes

#	Name/Description	Format
	1 - MCS ERS FS 2 - SPACETEC ERS HR FS 3 - IAI ERS HR FS 4 - LABEN ERS HR FS 5 - SPACETEC/ACS ERS FS 6 - ENERTEC MSS FS 7 - ENERTEC TM FS 8 - ACS SW FORMAT SYNCH 14 - ACS XSAR FORMAT SW SYNCH	
3	**Reserved** Reserved	xs:unsignedInt Min Occurs: 2 Max Occurs: 2 4 bytes

Table 20: Transcription_Id_Type Specification

3.2.1.1.2.5.Orbit_Acquisition_Id_Type

Orbit and Acquisition Identification

#	Name/Description	Format
1	**Track_Number** Track or Data Take Number (when applicable)	xs:integer 4 bytes
2	**Orbit** Orbit number (when applicable)	xs:integer 4 bytes
3	**Reserved_1** Reserved	xs:integer Min Occurs: 9 Max Occurs: 9 4 bytes
4	**Reserved_2** Reserved	xs:short Min Occurs: 3 Max Occurs: 3 2 bytes
5	**Reserved_3** Reserved	xs:hexBinary 18bytes
6	**Transcription_Date** Transcription Date in D M Y (WARNING: Year could be expressed in some tapes as years from 1900)	xs:short Min Occurs: 3 Max Occurs: 3 2 bytes
7	**Reserved_4** Reserved	xs:short Min Occurs: 6 Max Occurs: 6 2 bytes
8	**Filler** Padding for stucture alignment	xs:hexBinary 2bytes

Table 21: Orbit_Acquisition_Id_Type Specification

3.2.1.1.2.6.Pointers_Tape_Data_Type

Pointers to Tape Data Structure

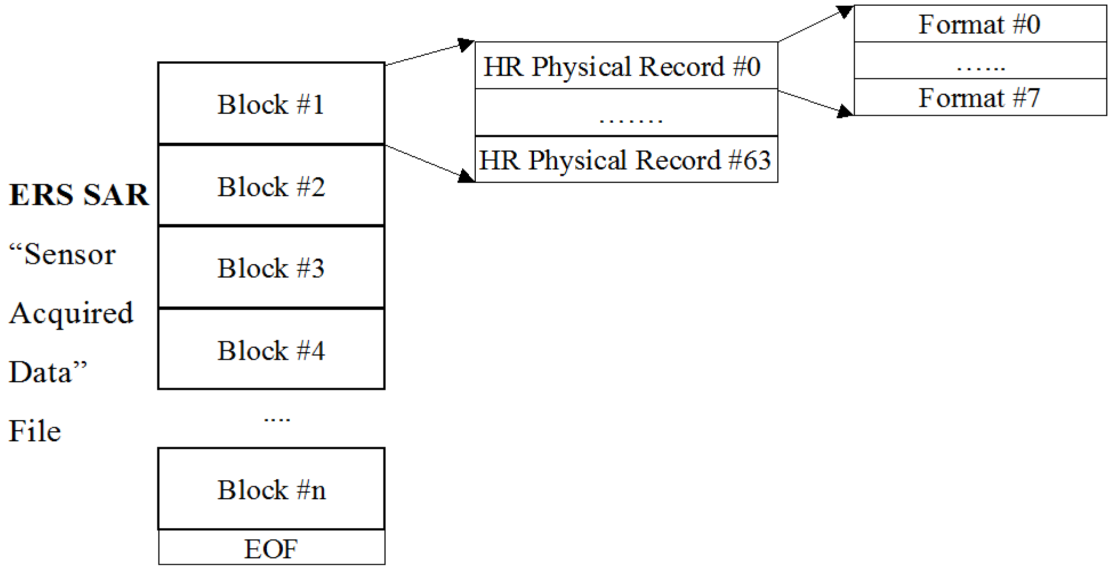
#	Name/Description	Format
1	**Reserved_1** Reserved	xs:integer Min Occurs: 5 Max Occurs: 5 4 bytes
2	**Physical_Address_1** User Header Address. File Number of the User Header. It is the number of files, numbered from DLT start, preceding the User Header file in current Transcription Area.	xs:long 4 bytes
3	**Physical_Address_2** Pass Id. Header Address. File Number of the Pass Id. Header. It is the number of files, numbered from DLT start, preceding the Pass Id. Header file in current Transcription Area.	xs:long 4 bytes
4	**Reserved_2** Reserved	xs:hexBinary 652bytes

Table 22: Pointers_Tape_Data_Type Specification

3.2.1.2. DTVideoData

The ERS SAR Sensor data of one imaging sequence are stored consecutively in the DTVideoData file. ERS Sensor Data are stored in fixed length records, each one containing a fixed number of “formats”.

A Block is formed by 64 physical records, each record containing 8 “formats”.



Each “format” is decomposed in 29 frames:

- 1 frame, with auxiliary data.
- 28 frames with the measurement data.

The next figure provides a high level overview of the complex structures used to represent the information of the DTVideoData file:

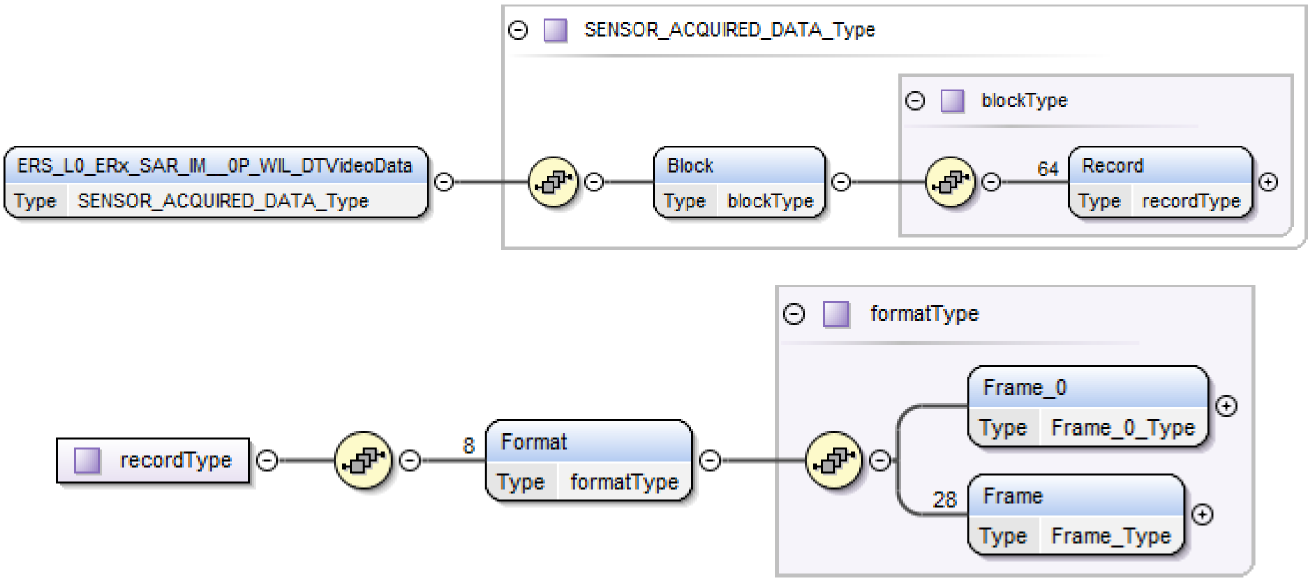


Figure 2: DFDL schema organisation for SAR_IM__0P product in WILMA format - DTVideoData

3.2.1.2.1. Root Element

#	Name/Description	Format
1	**ERS_L0_ERx_SAR_IM__0P_WIL_DTVideoData** ERS SAR IM Level-0 Product File in WILMA Format SENSOR ACQUIRED DATA FILE The Video Data are transcribed into tape as they are received in the memory of the host computer. Their format depends on the specific type of sensor. There is some dependence in the structure of transcribed data block on the Format Synchroniser that has been used for the ingestion of the data into the computer memory. In fact, Format Synchronisers do not manipulate the data, but some reformatting is sometimes executed. This implies that two transcription files of the same pass, made by Transcription systems that use different Format Synchronisers, may slightly differ in marginal aspects. This is the reason why we put, in the fields of the Header files of the Transcription Area, the code of the Format Synchroniser used for the transcription. The reproduction software must take care of this information to manage the transcribed data. Following paragraphs detail the various cases.	SENSOR_ACQUIRED_DATA_Type

#	Name/Description	Format
	The ERS SAR Sensor Data of one imaging sequence are stored consecutively in the file. The characteristics and the quality level of each segment are contained in the "Segment Descriptor" file.	

Table 23: ERS_L0_ERx_SAR_IM__0P_WIL_DTVideoData Specification

3.2.1.2.2. Complex Types

3.2.1.2.2.1.SENSOR_ACQUIRED_DATA_Type

#	Name/Description	Format
1	**Block**	blockType Min Occurs: 1 Max Occurs: 1

Table 24: SENSOR_ACQUIRED_DATA_Type Specification

3.2.1.2.2.2.blockType

A Block is formed (not physically but logically) by 64 physical records and has the usual length of about 4 Megabytes: (59392*64 = 3,801,088 bytes).

#	Name/Description	Format
1	**Record**	recordType Min Occurs: 64 Max Occurs: 64

Table 25: blockType Specification

3.2.1.2.2.3.recordType

ERS Sensor Data are stored in fixed length records, each containing a fixed number of formats. The fixed length record is the tape physical record of the Sensor Acquired Data file. The number of formats per record is 8. The size of the physical record on tape is therefore: (29*256)*8 = 59392 bytes.

#	Name/Description	Format
1	**Format**	formatType Min Occurs: 8 Max Occurs: 8

Table 26: recordType Specification

3.2.1.2.2.4.formatType

One High Rate Format consists of 29 frames of 256 bytes each. The data of each format are generated by two different sources: * The IDHT, which generates the synchronization code, the IDHT identification code and the IDHT General Header. * The AMI, which generates data contained in the auxiliary data field and the AMI Measurement data field.

#	Name/Description	Format
1	**Frame_0**	Frame_0_Type
2	**Frame**	Frame_Type

#	Name/Description	Format
		Min Occurs: 28 Max Occurs: 28

Table 27: formatType Specification

3.2.1.2.2.5.Frame_0_Type

#	Name/Description	Format
1	**IDHT_SyncCode** IDHT Synchronisation Code.	xs:hexBinary 3 bytes
2	**IDHT_ID** IDHT Identification Code	xs:hexBinary 3 bytes
3	**IDHT_General_Header** IDHT General Header	IDHT_General_Header_Type
4	**auxiliaryData** Auxiliary Data and Calibration or Replica Data (Array of 220 unsignedBytes).	xs:hexBinary 220 bytes
5	**measurementData** Echo or noise data are named measurement data (Array of 20 unsignedBytes).	xs:hexBinary 20 bytes

Table 28: Frame_0_Type Specification

3.2.1.2.2.6.IDHT_General_Header_Type

#	Name/Description	Format
1	**Packet_counter** Packet counter	xs:hexBinary 1 bytes
2	**Subcommutation_counter** Subcommutation counter	xs:hexBinary 1 bytes
3	**IDHT_General_Header_source_packet** IDHT General Header source packet	xs:hexBinary 8 bytes

Table 29: IDHT_General_Header_Type Specification

3.2.1.2.2.7.Frame_Type

#	Name/Description	Format
1	**IDHT_SyncCode** IDHT Synchronisation Code	xs:hexBinary 3 bytes
2	**IDHT_ID** IDHT Identification Code	xs:hexBinary 3 bytes
3	**measurementData** Echo or noise data are named measurement data (Array of 250 unsignedBytes).	xs:hexBinary 250 bytes

Table 30: Frame_Type Specification

3.2.1.3. DTUserHeader

The DTUserHeader file contains the acquisition description as well as the logical and physical file structure. It contains all the parameters of the imaging sequence, the orbital parameters, the information about acquisition station, the description of the file structure and contents and all information necessary for further processing. It has the same structure for all satellites transcriptions.

The next figure provides a high level overview of the complex structures used to represent the information of the DTUserHeader file:

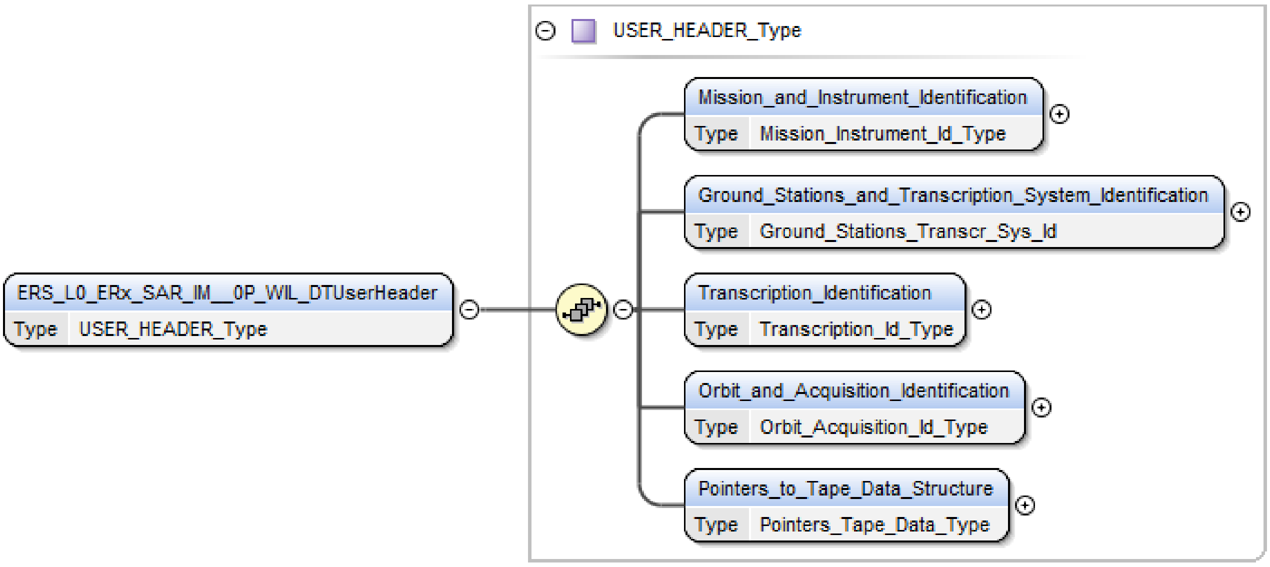

USER_HEADER_Type
ERS_L0_ERx_SAR_IM__0P_WIL_DTUserHeader
Type USER_HEADER_Type
Mission_and_Instrument_Identification
Type Mission_Instrument_Id_Type
Ground_Stations_and_Transcription_System_Identification
Type Ground_Stations_Transcr_Sys_Id
Transcription_Identification
Type Transcription_Id_Type
Orbit_and_Acquisition_Identification
Type Orbit_Acquisition_Id_Type
Pointers_to_Tape_Data_Structure
Type Pointers_Tape_Data_Type

Figure 3: DFDL schema organisation for SAR_IM__0P product in WILMA format - DTUserHeader

3.2.1.3.1. Root Element

#	Name/Description	Format
1	**ERS_L0_ERx_SAR_IM__0P_WIL_DTUserHeader** ERS SAR IM Level-0 Product File in WILMA Format USER HEADER file The "User Header" file contains the acquisition description as well as the logical and physical file structure. It contains all the parameters of the imaging sequence, the orbital parameters, the information about acquisition station, the description of the file structure and contents and all information necessary for further processing. It has the same structure for all satellites transcriptions. The file is divided into five logical sections: 1. Mission and Instrument Identification 2. Ground Stations and Transcription System Identification 3. Transcription Identification 4. Orbit and Acquisition Identification 5. Pointers to Tape Data Structure	USER_HEADER_Type

Table 31: ERS_L0_ERx_SAR_IM__0P_WIL_DTUserHeader Specification

3.2.1.3.2. Complex Types

3.2.1.3.2.1.USER_HEADER_Type

#	Name/Description	Format
1	**Mission_and_Instrument_Identification**	Mission_Instrument_Id_Type
2	**Ground_Stations_and_Transcription_System_Identification**	Ground_Stations_Transcr_Sys_Id
3	**Transcription_Identification**	Transcription_Id_Type
4	**Orbit_and_Acquisition_Identification**	Orbit_Acquisition_Id_Type
5	**Pointers_to_Tape_Data_Structure**	Pointers_Tape_Data_Type

Table 32: USER_HEADER_Type Specification

3.2.1.3.2.2.Mission_Instrument_Id_Type

Mission and Instrument Identification

#	Name/Description	Format
1	**Reserved_1** Reserved	xs:hexBinary 76 bytes
2	**Satellite_ID** Satellite Code: 1 - LANDSAT 2 - MOS 3 - J-ERS 4 - SPOT 5 - ERS 6 - IRS-C 7 - RADARSAT 8 - NOAA 9 - RESERVED 10 - RESERVED 11 - HELIOS 12 - SHUTTLE 13 - EOSAM 14 - EOSPM	xs:short 2 bytes
3	**Mission_ID** Satellite Mission Number	xs:short 2 bytes
4	**Instr_Type_ID** Satellite Mission Number: 1 - LANDSAT MSS 2 - LANDSAT TM 3 - LANDSAT ETM 4 - LANDSAT RBV 5 - MOS MESSR 6 - J-ERS VNIR 7 - J-ERS SWIR 8 - Not Used 9 - Not Used	xs:short 2 bytes

#	Name/Description	Format
	10 - ERS AMI SAR 11 - ERS ATSR 12 - SPOT HRV 13 - J-ERS SAR 14 - NOAA AVHRR 15 - SPOT HRVIR 16 - SHUTTLE XSAR 17 - MODIS	
5	**Instr_Number** Instrument number (when applicable)	xs:short 2 bytes
6	**Transm_Channel** Instrument number (when applicable)	xs:short 2 bytes

Table 33: Mission_Instrument_Id_Type Specification

3.2.1.3.2.3.Ground_Stations_Transcr_Sys_Id

Ground Stations and Transcription System Identification

#	Name/Description	Format
1	**Station_ID** Acquisition Ground Station Code: 67 - Adelaide 97 - Agrhymet 10 - AliceSpring 68 - Aspendale 105 - Atlanta 20 - Aussaguel 6 - Bangkok 52 - BantonRouge 49 - Bedford 74 - Beijing 30 - Berlin 31 - Berne 112 - Bishkek 29 - Bremenhaven 28 - Budapest 96 - Cairo 92 - Casey 64 - Cashoiera 32 - Copenhagen 103 - Cordoba 8 - Cotopaxi 23 - Cuiaba 76 - Da-Xing 33 - DeBilt 75 - Dhaka 50 - Downsview 34 - Dundee 51 - Edmonton 7 - Fairbanks	xs:short 2 bytes

#	Name/Description	Format
	24 - Farnborough 35 - Frascati 1 - Fucino 65 - Funceme 9 - Gatineau 54 - GilmoreCreek 36 - Hamburg 19 - HarteBeesHoek 16 - Hatoyama 37 - Helsinki 22 - Hobart 69 - Honolulu 11 - Hyderabad 21 - Islamabad 115 - Itu 77 - Keelung 116 - KhantyMansiysk 2 - Kiruna 78 - Kiyose 113 - Kitab 111 - Kourou 38 - Krakow 91 - KualaLumpur 12 - Kumamoto 39 - Lannion 79 - Lapan 99 - LaReunion 40 - Lasham 106 - Libreville 41 - Madrid 109 - Malindi 80 - Manila 3 - Maspalomas 104 - Matera 93 - McMurdo 53 - Miami 110 - Moscow 98 - Nairobi 55 - NESDIS 108 - Neustrelitz 81 - NewDelhi 102 - Norman 42 - Norrkoping 27 - Oberpfaff 43 - Offenbach 15 - OHiggins 44 - Oslo 94 - Palmer 18 - PariPari 70 - Perth	

#	Name/Description	Format
	45 - Prague 25 - Pretoria 13 - PrinceAlbert 57 - RedwoodCity 82 - Riyadh 46 - Rome 101 - RRSC_Nairobi 66 - Santiago 26 - Scanzano 58 - Scipps Inst 84 - Selangor 87 - Sendai 85 - Seoul 86 - SeoulUniv 83 - Singapore 59 - SiouxFalls 47 - Spitzenbergen 60 - StennisSpace 71 - Sydney 17 - Syowa 90 - Taipei 5 - Tel_Aviv 95 - TerranovaBay 88 - TokaiUniv 89 - TokyoUniv 72 - Townsville 48 - Traben-Trar 4 - Tromso 100 - Tunis 114 - UlanBator 61 - UnivOfAlaska 63 - UnivOfRhodeIsl 62 - UnivOfTexas 56 - WallopsIsl 73 - Wellington 14 - WestFreugh	
2	**Station_DT_ID** Data Transcription Ground Station Code	xs:short 2 bytes
3	**Filler** Padding for stucture alignment	xs:hexBinary 2 bytes

Table 34: Ground_Stations_Transcr_Sys_Id Specification

3.2.1.3.2.4.Transcription_Id_Type

Transcription Identification

#	Name/Description	Format
1	**Input_HddrType** HDDR Code	xs:unsignedInt 4 bytes
2	**Format_SyncType**	xs:unsignedInt

#	Name/Description	Format
	Format Synchroniser/Decommutator Code: 1 - MCS ERS FS 2 - SPACETEC ERS HR FS 3 - IAI ERS HR FS 4 - LABEN ERS HR FS 5 - SPACETEC/ACS ERS FS 6 - ENERTEC MSS FS 7 - ENERTEC TM FS 8 - ACS SW FORMAT SYNCH 14 - ACS XSAR FORMAT SW SYNCH	4 bytes
3	**Reserved** Reserved	xs:unsignedInt Min Occurs: 2 Max Occurs: 2 4 bytes

Table 35: Transcription_Id_Type Specification

3.2.1.3.2.5.Orbit_Acquisition_Id_Type

Orbit and Acquisition Identification

#	Name/Description	Format
1	**Track_Number** Track or Data Take Number (when applicable)	xs:integer 4 bytes
2	**Orbit_Number** Orbit number (when applicable)	xs:integer 4 bytes
3	**Cycle_Number** Cycle number (when applicable)	xs:integer 4 bytes
4	**Numb_of_Frames** Number of standard frames (when applicable)	xs:integer 4 bytes
5	**First_Frame** Num. of first standard frame (when applicable)	xs:integer 4 bytes
6	**Reserved** Reserved	xs:unsignedInt Min Occurs: 4 Max Occurs: 4 4 bytes
7	**First_OBC** 1st On Board Counter (when applicable)	xs:unsignedInt 4 bytes
8	**Last_OBC** Last On Board Counter (when applicable)	xs:unsignedInt 4 bytes
9	**Acquis_Date_Year** Acquisition Date (Year)	xs:unsignedShort 2 bytes
10	**Acquis_Date_Month** Acquisition Date (Month)	xs:unsignedShort 2 bytes
11	**Acquis_Date_Day** Acquisition Date (Day)	xs:unsignedShort 2 bytes
12	**Acquis_Day** Day in the year of the acquisition	xs:unsignedShort 2 bytes
13	**Acquis_start_Hour** Start of acquisition (Hour)	xs:unsignedShort 2 bytes

#	Name/Description	Format
14	**Acquis_start_Min** Start of acquisition (Minutes)	xs:unsignedShort 2 bytes
15	**Acquis_start_Sec** Start of acquisition (Seconds)	xs:unsignedShort 2 bytes
16	**Acquis_start_Millisec** Start of acquisition (Milliseconds)	xs:unsignedShort 2 bytes
17	**Acquis_end_Hour** End of acquisition (Hours)	xs:unsignedShort 2 bytes
18	**Acquis_end_Min** End of acquisition (Minutes)	xs:unsignedShort 2 bytes
19	**Acquis_end_Sec** End of acquisition (Seconds)	xs:unsignedShort 2 bytes
20	**Acquis_end_Millisec** End of acquisition (Milliseconds)	xs:unsignedShort 2 bytes
21	**Transcription_Date_Day** Transcription Date in Days	xs:unsignedShort 2 bytes
22	**Transcription_Date_Month** Transcription Date in Months	xs:unsignedShort 2 bytes
23	**Transcription_Date_Year** Transcription Date in Years (WARNING: Year could be expressed in some tapes as years from 1900)	xs:unsignedShort 2 bytes
24	**Transcription_Start_Hour** Transcription start in Hours	xs:unsignedShort 2 bytes
25	**Transcription_Start_Min** Transcription start in Minutes	xs:unsignedShort 2 bytes
26	**Transcription_Start_Sec** Transcription start in Seconds	xs:unsignedShort 2 bytes
27	**Transcription_End_Hour** Transcription end in Hours	xs:unsignedShort 2 bytes
28	**Transcription_End_Min** Transcription end in Minutes	xs:unsignedShort 2 bytes
29	**Transcription_End_Sec** Transcription end in Seconds	xs:unsignedShort 2 bytes
30	**Filler** Padding for stucture alignment	xs:hexBinary 2 bytes

Table 36: Orbit_Acquisition_Id_Type Specification

3.2.1.3.2.6.Pointers_Tape_Data_Type

Pointers to Tape Data Structure This structure is meaningful only in the "User Header" file (it is empty in "Pass Id. Header").

The last field of the first area (Nr_of_Files) indicates how many files follow the "User Header" inside the same Transcription Area (same passage). This number has been taken as a parameter to allow the maximum flexibility. In the Transcription Systems presently designed and installed the parameter "Nr_of_Files" can be 2 (LANDSAT), 3 (ERS SAR, SPOT HVR and HRVIR) or 4 (J-ERS SAR).

Nevertheless in the future other satellites could have different data structure and "Nr_of Files" could vary correspondingly up to 10. The 10 blocks which follow "Nr_of Files" describes the files following the "User Header" in current "Transcription Area". Each of these blocks describes one file.

The number of blocks actually filled is thus equal to "Nr_of_Files" In ERS SAR, SPOT (HRV and HRVIR) and SHUTTLE XSAR and case there are 3 blocks filled, which correspond respectively to: 1. the "Segment Descriptor" file, 2. the "Orbit Data" file, 3. the "Block Addresses Descriptor" file.

#	Name/Description	Format
1	**Num_of_segments** Number of segments (when applicable)	xs:integer 4 bytes
2	**Loaded_Swath** Number of transcribed swaths	xs:integer 4 bytes
3	**Swath_Size** Swath length (in bytes)	xs:integer 4 bytes
4	**Swath_per_Block** Number of swaths per block	xs:integer 4 bytes
5	**Nr_of_Blocks** Number of blocks	xs:integer 4 bytes
6	**Physical_Address_1** User Header Address. File Number of the User Header. It is the number of files, numbered from DLT start, preceding the User Header file in current Transcription Area.	xs:long 4 bytes
7	**Physical_Address_2** Pass Id. Header Address. File Number of the Pass Id. Header. It is the number of files, numbered from DLT start, preceding the Pass Id. Header file in current Transcription Area.	xs:long 4 bytes
8	**Nr_of_Files** Number of files following the present	xs:integer 4 bytes
9	**File**	Pointers_Tape_Data_File_Type Min Occurs: 10 Max Occurs: 10
10	**Reserved** Reserved	xs:hexBinary 8bytes

Table 37: Pointers_Tape_Data_Type Specification

3.2.1.3.2.7.Pointers_Tape_Data_File_Type

#	Name/Description	Format
1	**File_ID** File type identifier	xs:integer 4 bytes
2	**File_Num** Number of physical records	xs:integer 4 bytes
3	**Record_Length** Physical record length in bytes	xs:integer 4 bytes
4	**Elem_Num** Number of logical element per record	xs:integer 4 bytes
5	**Elem_Length** Logical element length in bytes	xs:integer 4 bytes

#	Name/Description	Format
6	**Filler** Spare	xs:hexBinary 44 bytes

Table 38: Pointers_Tape_Data_File_Type Specification

3.2.1.4. DTSegment

This DTSegment file contains the descriptions of all the segments in which the satellite pass has been divided.

The next figure provides a high level overview of the complex structures used to represent the information of the DTSegment file:

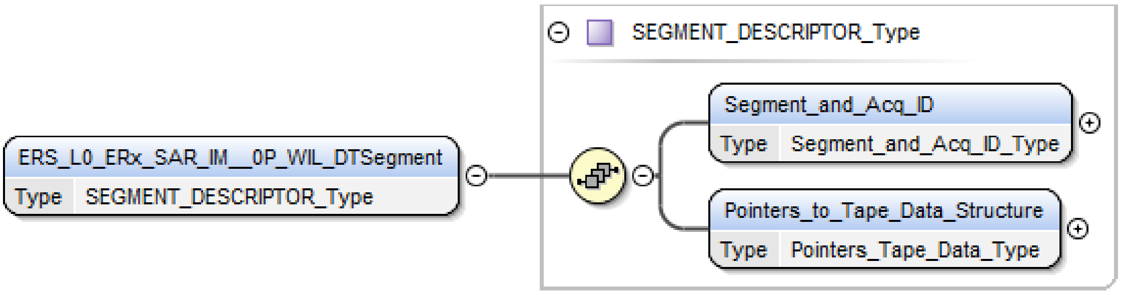

SEGMENT_DESCRIPTOR_Type
ERS_L0_ERx_SAR_IM__0P_WIL_DTSegment
Type SEGMENT_DESCRIPTOR_Type
Segment_and_Acq_ID
Type Segment_and_Acq_ID_Type
Pointers_to_Tape_Data_Structure
Type Pointers_Tape_Data_Type

Figure 4: DFDL schema organisation for SAR_IM__0P product in WILMA format - DTSegment

3.2.1.4.1. Root Element

#	Name/Description	Format
1	**ERS_L0_ERx_SAR_IM__0P_WIL_DTSegment** ERS SAR IM Level-0 Product File in WILMA Format SEGMENT DESCRIPTOR file This file contains the descriptions of all the segments in which the satellite pass has been divided. Each segment is described by one structure. The "Segment Descriptor Structure" is thus repeated as many times as the number of segments recorded in the "Sensor Acquired Data" file. Each segment is completely described by the following fields and contains all information to address it within the video data records. The number of items and the length are specified into the "User Header file - Pointer to Tape Data Structure Description Section" (described in previous paragraph).	SEGMENT_DESCRIPTOR_Type

Table 39: ERS_L0_ERx_SAR_IM__0P_WIL_DTSegment Specification

3.2.1.4.2. Complex Types

3.2.1.4.2.1.SEGMENT_DESCRIPTOR_Type

#	Name/Description	Format
1	**Segment_and_Acq_ID**	Segment_and_Acq_ID_Type
2	**Pointers_to_Tape_Data_Structure**	Pointers_Tape_Data_Type

Table 40: SEGMENT_DESCRIPTOR_Type Specification

3.2.1.4.2.2.Segment_and_Acq_ID_Type

Segment and Acquisition Identification

#	Name/Description	Format
1	**Acquis_Date_Year** Acquisition Date of the Sat. Pass (Year)	xs:short 2 bytes
2	**Acquis_Date_Month** Acquisition Date of the Sat. Pass (Month)	xs:short 2 bytes
3	**Acquis_Date_Day** Acquisition Date of the Sat. Pass (Day)	xs:short 2 bytes
4	**Acquis_Day** Day in the year of the acquisition	xs:short 2 bytes
5	**Segment_Start_Hours** Start of Segment (Hours)	xs:short 2 bytes
6	**Segment_Start_Min** Start of Segment (Min)	xs:short 2 bytes
7	**Segment_Start_Sec** Start of Segment (Sec)	xs:short 2 bytes
8	**Segment_Start_Millisec** Start of Segment (Millisec)	xs:short 2 bytes
9	**Segment_End_Hours** End of Segment (Hours)	xs:short 2 bytes
10	**Segment_End_Min** End of Segment (Min)	xs:short 2 bytes
11	**Segment_End_Sec** End of Segment (Sec)	xs:short 2 bytes
12	**Segment_End_Millisec** End of Segment (Millisec)	xs:short 2 bytes
13	**Loaded_Swath** Nr. of lines loaded on tape for this segment	xs:integer 4 bytes
14	**First_Swath** First swath of the segment	xs:integer 4 bytes
15	**Last_Swath** Last swath of the segment	xs:integer 4 bytes
16	**Lost_Swath** Lost swaths of the segment	xs:integer 4 bytes
17	**First_Frame** First frame of the segment (when applicable)	xs:integer 4 bytes

#	Name/Description	Format
18	**Last_Frame** Last frame of the segment (when applicable)	xs:integer 4 bytes
19	**First_OBC** First On Board Counter or TSID (when applicable)	xs:integer 4 bytes
20	**Last_OBC** Last On Board Counter or TSID (when applicable)	xs:integer 4 bytes

Table 41: Segment_and_Acq_ID_Type Specification

3.2.1.4.2.3.Pointers_Tape_Data_Type

Pointers to Tape Data Structure

#	Name/Description	Format
1	**Starting_Address** Pass Id. Header file Address	xs:integer 4 bytes
2	**Swath_Size** Swath length (in bytes) This field is filled only if its value changed between different Segments. If it does not change the valid value for this field must be read in the User Header (Pointers to Tape Data Structure)	xs:integer 4 bytes
3	**Swath_per_Block** Number of swaths per block This field is filled only if its value changed between different Segments. If it does not change the valid value for this field must be read in the User Header (Pointers to Tape Data Structure)	xs:integer 4 bytes
4	**Nr_of_Blocks** Number of blocks This field is filled only if its value changed between different Segments. If it does not change the valid value for this field must be read in the User Header (Pointers to Tape Data Structure)	xs:integer 4 bytes
5	**Formats_Per_Swath** Number of formats per swath This field is filled only if its value changed between different Segments. If it does not change the valid value for this field must be read in the User Header (Pointers to Tape Data Structure)	xs:long 4 bytes
6	**Filler** Filler	xs:hexBinary 52bytes

Table 42: Pointers_Tape_Data_Type Specification

3.2.1.5. DTOrbitFile

The DTOrbitFile contains information about position, velocity of satellite and times closest to start acquisition. There is only one “Orbital Data Record” for each pass. The State Vector is referenced to the inertial co-ordinate system called ECITOD (Earth Centred Inertial True Of Date).

The next figure provides a high level overview of the complex structures used to represent the information of the DTOrbitFile:

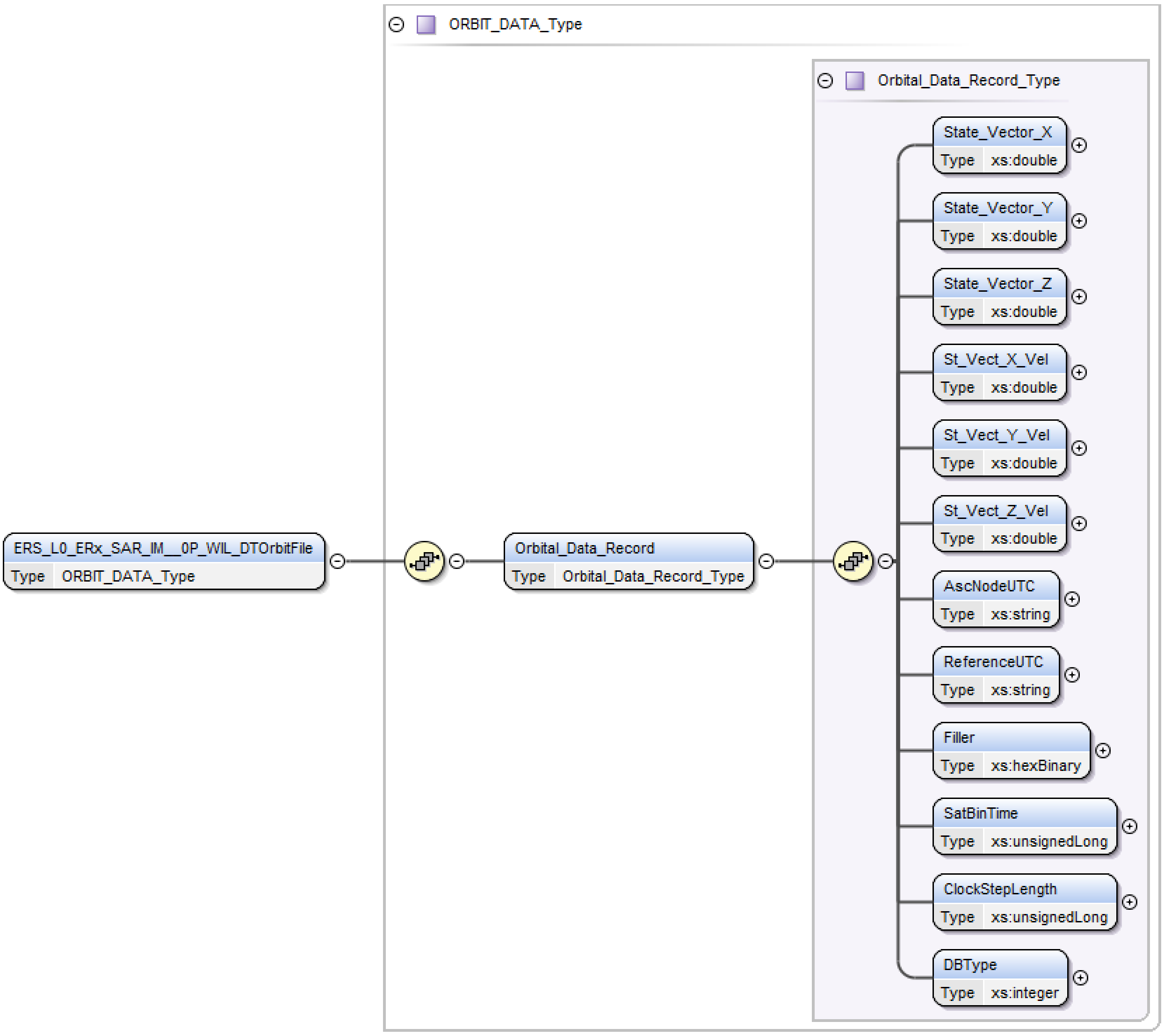


Figure 5: DFDL schema organisation for SAR_IM__0P product in WILMA format - DTOrbitFile

3.2.1.5.1. Root Element

#	Name/Description	Format
1	**ERS_L0_ERx_SAR_IM__0P_WIL_DTOrbitFile** ERS SAR IM Level-0 Product File in WILMA Format ORBIT DATA file This structure is composed by a series of blocks containing information about position, velocity of satellite and times closest to start acquisition. In the ERS case there is only one "Orbital Data	ORBIT_DATA_Type

#	Name/Description	Format
	Record" for each pass, i.e. for each Transcription Area. The State Vector is referenced to the inertial co-ordinate system called ECITOD (Earth Centred Inertial True Of Date).	

Table 43: ERS_L0_ERx_SAR_IM__0P_WIL_DTOrbitFile Specification

3.2.1.5.2. Complex Types

3.2.1.5.2.1.ORBIT_DATA_Type

#	Name/Description	Format
1	**Orbital_Data_Record**	Orbital_Data_Record_Type

Table 44: ORBIT_DATA_Type Specification

3.2.1.5.2.2.Orbital_Data_Record_Type

#	Name/Description	Format
1	**State_Vector_X** X component of satellite position	xs:double 8 bytes
2	**State_Vector_Y** Y component of satellite position	xs:double 8 bytes
3	**State_Vector_Z** Z component of satellite position	xs:double 8 bytes
4	**St_Vect_X_Vel** Vx component of sat. velocity	xs:double 8 bytes
5	**St_Vect_Y_Vel** Vy component of sat. velocity	xs:double 8 bytes
6	**St_Vect_Z_Vel** Vz component of sat. velocity	xs:double 8 bytes
7	**AscNodeUTC** UTC epoch of sat. position	xs:string 25characters
8	**ReferenceUTC** UTC reference time	xs:string 25characters
9	**Filler** Padding for structure alignment	xs:hexBinary 2bytes
10	**SatBinTime** Satellite reference binary time	xs:unsignedLong 4 bytes
11	**ClockStepLength** Satellite reference binary time	xs:unsignedLong 4 bytes
12	**DBType** 0 = predicted, 1 = restituted	xs:integer 4 bytes

Table 45: Orbital_Data_Record_Type Specification

3.2.1.6. DTBlock

This file contains the description of all the blocks in which the pass has been divided and written. It is composed by a variable number of identical units. Each unit describes completely one satellite data block and contains all information to address any segment or any frame within the “Sensor Acquired Data” file. It contains block number of the video data block, the starting time of the first satellite format in the block and the number of swaths contained.

The structure is repeated as many times as the number of blocks recorded in the file. The number of items and the length are specified in the "DTUserHeader” file.

The next figure provides a high level overview of the complex structures used to represent the information of the the DTBlock file:

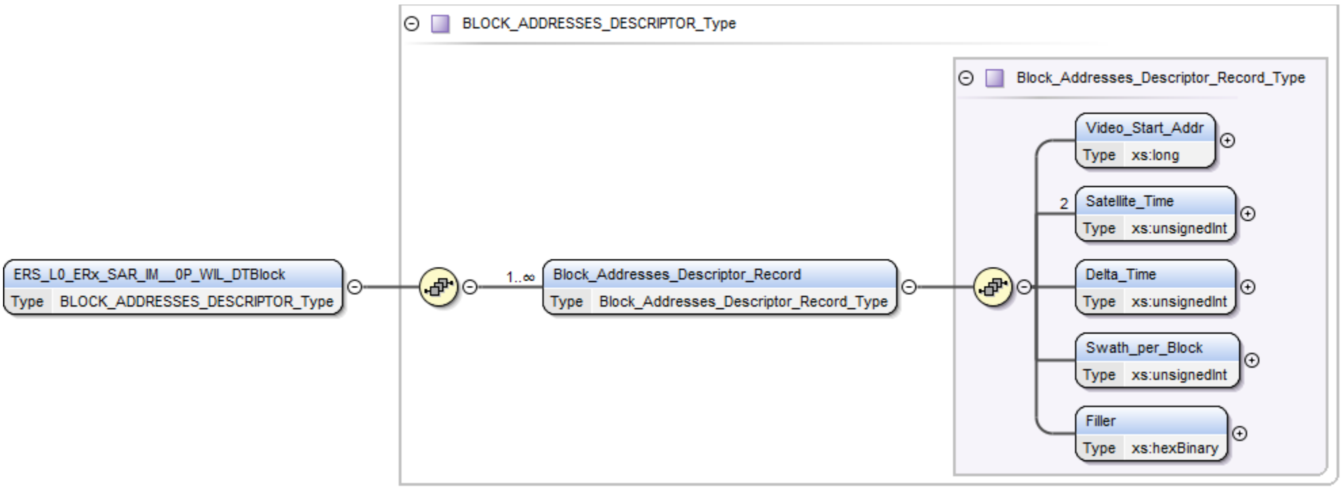

BLOCK_ADDRESSES_DESCRIPTOR_Type
Block_Addresses_Descriptor_Record_Type
Video_Start_Addr
Type xs:long
2
Satellite_Time
Type xs:unsignedInt
ERS_L0_ERx_SAR_IM__0P_WIL_DTBlock
Type BLOCK_ADDRESSES_DESCRIPTOR_Type
1..∞
Block_Addresses_Descriptor_Record
Type Block_Addresses_Descriptor_Record_Type
Delta_Time
Type xs:unsignedInt
Swath_per_Block
Type xs:unsignedInt
Filler
Type xs:hexBinary

Figure 6: DFDL schema organisation for SAR_IM__0P product in WILMA format - DTBlock

3.2.1.6.1. Root Element

#	Name/Description	Format
1	**ERS_L0_ERx_SAR_IM__0P_WIL_DTBlock** ERS SAR IM Level-0 Product File in WILMA Format BLOCK ADDRESSES DESCRIPTOR file This file contains the description of all the blocks in which the pass has been divided and written on tape. It is composed by a variable number of identical units. Each unit describes completely one satellite data block and contains all information to address any segment or any frame within the "Sensor Acquired Data" file. It contains block number of the video data block, the starting time of the first satellite format in the block and the number of swaths contained. The structure is repeated as many times as the number of blocks recorded in the file. The number of items and the length are specified into the "User Header file - Pointer to Tape Data	BLOCK_ADDRESSES_DESCRIPTOR_Type

#	Name/Description	Format
	Structure Description Section" (Block Addresses section).	

Table 46: ERS_L0_ERx_SAR_IM__0P_WIL_DTBlock Specification

3.2.1.6.2. Complex Types

3.2.1.6.2.1.BLOCK_ADDRESSES_DESCRIPTOR_Type

#	Name/Description	Format
1	**Block_Addresses_Descriptor_Record**	Block_Addresses_Descriptor_Record_Type Min Occurs : 1 Max Occurs : unbounded

Table 47: BLOCK_ADDRESSES_DESCRIPTOR_Type Specification

3.2.1.6.2.2.Block_Addresses_Descriptor_Record_Type

#	Name/Description	Format
1	**Video_Start_Addr** Block number of the video data block	xs:long 4 bytes
2	**Satellite_Time** Time of current block. The time is expressed in sensor dependent mode, as follows: Sat_Time[0] = Binary Counter of first line in current block Sat_Time[1] = Binary Counter of last line in current block	xs:unsignedInt Min Occurs : 2 Max Occurs : 2 4 bytes
3	**Delta_Time** Time distance between start of acquisition (first block of the file) and current block start The delta time is expressed in sensor dependent mode, as "Format counter of first format in current block"	xs:unsignedInt 4 bytes
4	**Swath_per_Block** Number of swaths per block	xs:unsignedInt 4 bytes
5	**Filler** Available fields	xs:hexBinary 12bytes

Table 48: Block_Addresses_Descriptor_Record_Type Specification

3.2.1.7. DTStatisticFile

The Statistics file constitutes a sort of summary of all transcribed passes.

The next figure provides a high level overview of the complex structures used to represent the information of the DTStatisticFile:

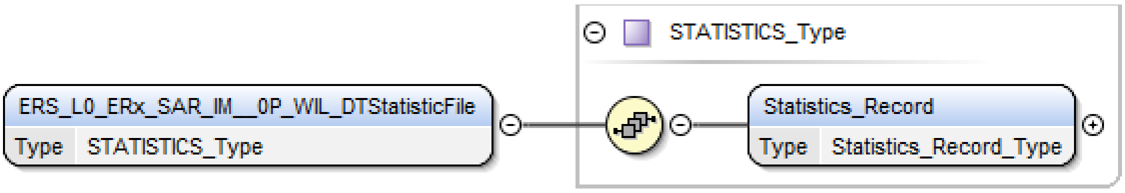

STATISTICS_Type
ERS_L0_ERx_SAR_IM__0P_WIL_DTStatisticFile
Type STATISTICS_Type
Statistics_Record
Type Statistics_Record_Type

Figure 7: DFDL schema organisation for SAR_IM__0P product in WILMA format - DTStatisticFile

3.2.1.7.1. Root Element

#	Name/Description	Format
1	**ERS_L0_ERx_SAR_IM__0P_WIL_DTStatisticFile** ERS SAR IM Level-0 Product File in WILMA Format STATISTICS file The Statistics file always follows the Transcription Area in the DLT and constitutes a sort of summary of the passes contained in the DLT. The Statistics File following the last Transcription Area of the cassette constitutes a directory, with information on all the transcribed passes. This structure allows an easy positioning of the tape on the requested pass. To achieve this goal, when the tape inspection is performed, the end of tape is reached without reading any data. Then the control jumps one file back (at the beginning of the last Statistics file). Reading this file the content of the whole DLT can be known. The first record is empty. The successive records are structured according the table below; each of them points to the successive Transcription Area stored on the cassette.	STATISTICS_Type

Table 49: ERS_L0_ERx_SAR_IM__0P_WIL_DTStatisticFile Specification

3.2.1.7.2. Complex Types

3.2.1.7.2.1.STATISTICS_Type

#	Name/Description	Format
1	**Statistics_Record**	Statistics_Record_Type

Table 50: STATISTICS_Type Specification

3.2.1.7.2.2.Statistics_Record_Type

#	Name/Description	Format
1	**Reserved_1** Reserved	xs:hexBinary 4bytes
2	**Satellite_ID**	xs:unsignedShort

#	Name/Description	Format
	Satellite Code	2 bytes
3	**Mission_ID** Mission. Number	xs:unsignedShort 2 bytes
4	**Instr_Type_ID** Instrument Type Code	xs:unsignedShort 2 bytes
5	**Reserved_2** Reserved	xs:unsignedShort Min Occurs: 2 Max Occurs: 2 2 bytes
6	**Station_ID** Acquisition Ground Station Code	xs:unsignedShort 2 bytes
7	**Reserved_3** Reserved	xs:short 2 bytes
8	**Filler** Padding for structure alignment	xs:hexBinary 2bytes
9	**Track_Number** Track Number (when applicable)	xs:integer 4 bytes
10	**Orbit_Number** Orbit number (when applicable)	xs:integer 4 bytes
11	**Reserved_4** Reserved	xs:integer 4 bytes
12	**Number_of_Frames** Number of standard frames (when applicable)	xs:integer 4 bytes
13	**First_Frame** Num of first standard frame (when applicable)	xs:integer 4 bytes
14	**Reserved_5** Reserved	xs:integer Min Occurs: 4 Max Occurs: 4 4 bytes
15	**Acquisition_Date_Year** Acquisition Date (Year)	xs:short 2 bytes
16	**Acquisition_Date_Month** Acquisition Date (Month)	xs:short 2 bytes
17	**Acquisition_Date_Day** Acquisition Date (Month)	xs:short 2 bytes
18	**Acquisition_Day** Acquisition Day of the year	xs:short 2 bytes
19	**Acquisition_Start_Hours** Start of acquisition (Hours)	xs:short 2 bytes
20	**Acquisition_Start_Min** Start of acquisition (Min)	xs:short 2 bytes
21	**Acquisition_Start_Sec** Start of acquisition (Sec)	xs:short 2 bytes
22	**Acquisition_Start_Millisec** Start of acquisition (Millisec)	xs:short 2 bytes
23	**Acquisition_End_Hours** End of acquisition (Hours)	xs:short 2 bytes
24	**Acquisition_End_Min** End of acquisition (Min)	xs:short 2 bytes

#	Name/Description	Format
25	**Acquisition_End_Sec** End of acquisition (Sec)	xs:short 2 bytes
26	**Acquisition_End_Millisec** End of acquisition (Millisec)	xs:short 2 bytes
27	**Transcription_Date_Day** Transcription Date (Day)	xs:short 2 bytes
28	**Transcription_Date_Month** Transcription Date (Month)	xs:short 2 bytes
29	**Transcription_Date_Year** Transcription Date (Year)(WARNING: Year could be expressed in some tapes as years from 1900)	xs:short 2 bytes
30	**Reserved_6** Reserved	xs:short Min Occurs: 6 Max Occurs: 6 2 bytes
31	**Filler** Padding for structure alignment	xs:hexBinary 2bytes
32	**Reserved_7** Reserved	xs:integer Min Occurs: 10 Max Occurs: 10 4 bytes
33	**Physical_Address_1** User Header file number	xs:long 4 bytes
34	**Physical_Address_2** Pass Id. Header file number	xs:long 4 bytes
35	**Reserved_8** Reserved	xs:hexBinary 697bytes
36	**Copy_Date_Day** Date when this record has been generated as a copy from another tape (Day)	xs:unsignedByte 1bytes
37	**Copy_Date_Month** Date when this record has been generated as a copy from another tape (Month)	xs:unsignedByte 1bytes
38	**Copy_Date_Year** Date when this record has been generated as a copy from another tape (Year)	xs:unsignedByte 1bytes
39	**Copy_Source_Tape_Number** Number of the source tape from where this record was generated	xs:long 4 bytes
40	**Copy_Flag** 1 = this is a copied record 0 = this is the original transcribed record	xs:unsignedByte 1bytes
41	**Copy_Source_Media_Type** 4 = DLT 2 = SONY ID1	xs:unsignedByte 1bytes
42	**Reserved_9** Reserved	xs:hexBinary 2bytes

Table 51: Statistics_Record_Type Specification

*3.2.2. SAR_EWA_0P (LRD*F format)*

The next figure provides a high level overview of the complex structures used to represent the information of the SAR_EWA_0P in LRD*F format:

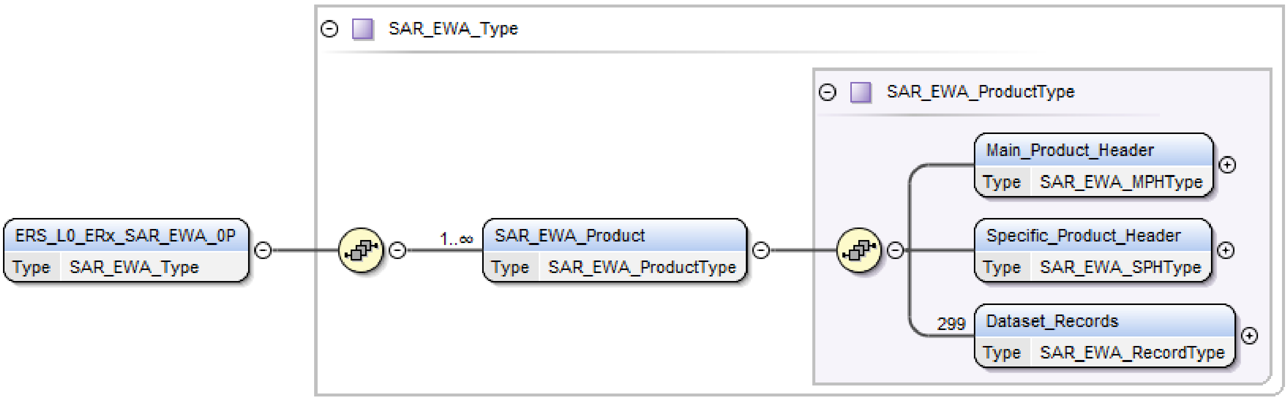


Figure 8: DFDL schema organisation for SAR_EWA_0P files

3.2.2.1. Root Element

#	Name/Description	Format
1	**ERS_L0_ERx_SAR_EWA_0P** This product contains 299 sets of platform data included in the Source Data Field of the IDHT General Header Source Packet. One product includes.: * Main Product Header * Specific Product Header * 299 Sensor Acquired Data records	SAR_EWA_Type

Table 52: ERS_L0_ERx_SAR_EWA_0P Specification

3.2.2.2. Complex Types

3.2.2.2.1. SAR_EWA_Type

#	Name/Description	Format
1	**SAR_EWA_Product**	SAR_EWA_ProductType Min Occurs : 1 Max Occurs : unbounded

Table 53: SAR_EWA_Type Specification

3.2.2.2.2. SAR_EWA_ProductType

#	Name/Description	Format
1	**Main_Product_Header**	SAR_EWA_MPHType
2	**Specific_Product_Header**	SAR_EWA_SPHType
3	**Dataset_Records**	SAR_EWA_RecordType Min Occurs : 299 Max Occurs : 299

Table 54: SAR_EWA_ProductType Specification

3.2.2.2.3. SAR_EWA_MPHType

#	Name/Description	Format
1	**Product_identifier** Product identifier (for ESA internal operational use only), i.e. a set of characters and integers which form a unique identifier	Product_identifierType
2	**Product_type** Type of Product: 0 RATSR ATSR-1 Extracted Calibration data 1 UI16 AMI Image 16-bit Fast delivery 2 UI8 AMI Image 8-bit Fast delivery 3 UIND AMI Image Noise Statistics and Drift Calibration 4 UIC AMI Image Chirp Replica 5 UWA AMI Wave Fast delivery 6 UWAND AMI Wave Noise Statistics and Drift Calibration 7 UWAC AMI Wave Chirp Replica 8 UWI AMI Wind Fast delivery 9 URA Radar Altimeter Fast delivery 10 IWA AMI Wave Intermediate Products 11 II16 AMI Image Intermediate Products 12 EIC AMI Image Extracted Calibration data Calibration Data 13 EWAC AMI Wave Extracted Calibration data Calibration Data 14 EWIC AMI Wind Extracted Calibration data Calibration Data 15 ERAC Radar Altimeter Extracted Calibration Data 16 EII AMI Image Instrument Headers 17 EWAI AMI Wave Instrument Headers 18 EWII AMI Wind Instrument Headers 19 ERAI Radar Altimeter Instrument Headers 20 EGH General Headers 21 EEP Ephemeris Data 22 TP Text Product 23 UILR User Image Low Resolution Image 30 VI Verification Image 31 VIC Verification Image Calibration 32 VWA Verification Wave 33 VWAC Verification Wave Calibration 34 EGOC GOME Extracted Calibration data 35 EGOI GOME Instrument Headers 36 EATI2 ATSR-2 Instrument Headers 37 EATI1 ATSR-1 Instrument Headers 38 EATC2 ATSR-2 Low rate Extracted Calibration data	xs:unsignedByte 1 bytes

#	Name/Description	Format
	39 EMWC Microwave Sounder Extracted Calibration data 40 EICM Multiple AMI Image Calibration Data	
3	**Spacecraft** Spacecraft 1: ERS-1 2: ERS-2	xs:unsignedByte 1 bytes
4	**Start_time** UTC time 7 of subsatellite point at beginning of product. Format in ASCII: DD-MMM-YYYY hh:mm:ss.ttt For example: 30-JAN-1987 14:30:27.123	xs:string 24characters
5	**Acquisition_station** Station ID, where data was processed 1: Kiruna Station (KS) 2: Fucino Station (FS) 3: Gatineau Station (GS) 4: Maspalomas Station (MS) 5: EECF Station (ES) 6: Prince Albert Station (PS)	xs:unsignedByte 1 bytes
6	**Product_Confidence_Data** Product Confidence Data bit 1 PCD Summary Flag 0: product correctly generated 1: at least one of the remaining 15 bits of the PCD in the MPH is set. In particular the specific header flags are not read when this bit is set. bit 2 - 3 spare bit 4 - 5 Downlink Performance and X-Band acquisition chain. This value summarizes the PCD snapshots rel. to the products. 0: performance better than MMCC/EECF-supplied minimum threshold 1: performance equal to or worse than threshold 2: performance unknown bit 6 - 7 HDDT Summary. This value summarizes the PCD snapshots rel. to the product. 0: performance better than MMCC/EECF-supplied minimum threshold 1: performance equal to or worse than threshold 2: performance unknown bit 8 - 9 Frame Synchronizer. This value summarizes the PCD snapshots rel. to the product. 0: performance better than MMCC/EECF-supplied minimum threshold 1: performance equal to or worse than threshold 2: performance unknown bit 10 - 11 FS to Processor I/F The LRDPF and SARFDP reads the status of the FS interface.	xs:unsignedShort 2 bytes

#	Name/Description	Format
	0: no parity error detected 1: at least one parity error detected 2: performance unknown bit 12 - 13 Checksum Analysis on LR Frames. The percentage of source packets, featuring a checksum error, and used in the actual product is compared to a MMCC/EECF given threshold. 0: lower than threshold 1: greater than threshold 2: performance unknown bit 14 - 15 Quality of Downlinked Formats and Source Packets. The RA product is based on using 80 consecutive source packets. The percentage of erroneous ones is determined and compared to a MMCC/EECF given threshold. 1: greater than threshold 2: performance unknown bit 16 Existence of Auxiliary Data (8For the LRDPF it has the meaning that at least one auxiliary data field in a source packet header is corrupted and for the SARFDP it means that the chirp could not be extracted from the input raw data. For UWI and URA this flag is always set to zero). 0: auxiliary data and/or chirp correctly extracted 1: not all auxiliary data extracted	
7	**Processing_Time** UTC time when MPH was generated	xs:string 24characters
8	**Size_of_SPH** Size of Specific Product Header: Record in Bytes	xs:unsignedInt 4 bytes
9	**Nb_of_PDS_Records** Number of Product Data Set Records	xs:unsignedInt 4 bytes
10	**Size_of_each_PDS_Record** Size of each Product Data Set Record in Bytes	xs:unsignedInt 4 bytes
11	**Subsystem** Subsystem that generated the product. 0: SARFDP 1 1: SARFDP 2 2: LRDPF 3: VMP 4: LRDTF	xs:unsignedByte 1 bytes
12	**OBRC_flag** OBRC flag used for SAR products only bit 1 - 2 0: not used 1: OGRC data 2: OBRC data	xs:unsignedByte 1 bytes
13	**UTC_ref_time** UTC reference time. Time relation used to convert	xs:string 24characters

#	Name/Description	Format
	from satellite to ground, used together with the next two fields.	
14	**Ref_bin_time_of_satellite_clock** Reference binary time of satellite clock (32-bit unsigned integer)	xs:unsignedInt 4 bytes
15	**Step_length_of_satellite_clock** Step length of satellite clock in nanoseconds	xs:unsignedInt 4 bytes
16	**Processor_software_version** Processor software version used to generate product.	xs:unsignedShort Min Occurs: 4 Max Occurs: 4 2 bytes
17	**Threshold_table_version** Threshold table version number.	xs:unsignedShort 2 bytes
18	**Spare** Spare	xs:hexBinary 2 bytes
19	**UTC_time_asc** UTC time of ascending node state vector	xs:string 24characters
20	**Asc_State_Vector** Ascending node state vector in earth-fixed reference system Product types which don't need valid orbit state vectors, e.g., UWAND and extracted products, for production, may contain invalid state vectors since these are not verified during production.	State_VectorType

Table 55: SAR_EWA_MPHType Specification

3.2.2.2.4. Product_identifierType

#	Name/Description	Format
1	**Originator_of_logical_schedule** Byte 1: Originator of logical schedule (for ESA internal use only) e.g.: I: MMCC/EECF, Immediate Command M: MMCC/EECF, Logical Schedule J: Local operator, Immediate Command K: Local operator, Logical Schedule	xs:string 1characters
2	**Sequential_Counter_of_logical_schedule** Byte 2-5: Sequential Counter of Logical Schedule	xs:unsignedInt 4 bytes
3	**Unique_Identification_or_Schedule_Offset** Byte 6-9: Unique Identification or Schedule Offset	xs:unsignedInt 4 bytes
4	**blank** Byte 10-13: Not used, set to 0	xs:integer 4 bytes

#	Name/Description	Format
5	**Sequential_Number_of_Currently_Generated_Product** Byte 14-17: Sequential Number of Currently Generated Product	xs:unsignedInt 4 bytes

Table 56: Product_identifierType Specification

3.2.2.2.5. State_VectorType

#	Name/Description	Format
1	**position** State vector; X, Y, Z position in 10-2 m	State_Vector_Coords
2	**velocity** State vector; X, Y, Z velocity in 10-5 m/s	State_Vector_Coords

Table 57: State_VectorType Specification

3.2.2.2.6. State_Vector_Coords

#	Name/Description	Format
1	**X**	xs:integer 4 bytes
2	**Y**	xs:integer 4 bytes
3	**Z**	xs:integer 4 bytes

Table 58: State_Vector_Coords Specification

3.2.2.2.7. SPHType

#	Name/Description	Format
1	**SPH_DESCRIPTOR="**	xs:string 16 bytes
	ASCII string describing the name of the Specific Product Header.	xs:string 28 bytes
	"	uc 1 bytes
2	**ProductLocationInformation**	ProductLocationInformationType
3	**ProductConfidenceDataInformation**	ProductConfidenceDataInformationType
4	**OtherProductConfidenceDataInformation**	OtherProductConfidenceDataInformationType
5	**ASARSpecificInformation**	ASARSpecificInformationType
6	**ProductDSD**	ProductDSDType Min Occurs : 3 Max Occurs : 3
7	**Spare**	xs:hexBinary 279bytes

Table 59: SPHType Specification

3.2.2.2.8. ProductLocationInformationType

#	Name/Description	Format
1	**START_LAT=**	xs:string 10 bytes
	Latitude of first satellite nadir point at the Sensing Start time (positive North) referred to WGS84: value<10-6degN>	xs:string 21 bytes
2	**START_LONG=**	xs:string 11 bytes
	Longitude of first satellite nadir point at the Sensing Start time (positive East, 0 = Greenwich) referred to WGS84: value<10-6degN>	xs:string 21 bytes
3	**STOP_LAT=**	xs:string 9 bytes
	Latitude of first satellite nadir point at the Sensing Stop time (positive North) referred to WGS84: value<10-6degN>	xs:string 21 bytes
4	**STOP_LONG=**	xs:string 10 bytes
	Longitude of first satellite nadir point at the Sensing Stop time (positive East, 0 = Greenwich) referred to WGS84: value<10-6degN>	xs:string 21 bytes
5	**SAT_TRACK=**	xs:string 10 bytes
	Sub-satellite track heading at the Sensing Start time in the MPH.	xs:string 20 bytes
6	**Spare**	xs:string 50 bytes

Table 60: ProductLocationInformationType Specification

3.2.2.2.8.1.ProductConfidenceDataInformationType

#	Name/Description	Format
1	**ISP_ERRORS_SIGNIFICANT=**	xs:string 23 bytes
	1 or 0. 1 if number of ISPs with CRC errors exceeds threshold For ERS, always set to a default value taken from PRM_Level0SphDefaults.dat parameter file.	uc
2	**MISSING_ISPS_SIGNIFICANT=**	xs:string 25 bytes
	1 or 0. 1 if number of missing ISPs exceeds threshold. For ERS, always set to a default value taken from PRM_Level0SphDefaults.dat parameter file.	uc

#	Name/Description	Format
3	**ISP_DISCARDED_SIGNIFICANT=**	xs:string 26 bytes
	1 or 0. 1 if number of ISPs discarded by the PF-HS exceeds threshold. For ERS, always set to a default value taken from PRM_Level0SphDefaults.dat parameter file.	uc
4	**RS_SIGNIFICANT=**	xs:string 15 bytes
	1 or 0. 1 if number of ISPs with Reed Solomon corrections exceeds threshold. For ERS, always set to a default value taken from PRM_Level0SphDefaults.dat parameter file.	uc
5	**Spare**	xs:string 50 bytes

Table 61: ProductConfidenceDataInformationType Specification

3.2.2.2.9. OtherProductConfidenceDataInformationType

#	Name/Description	Format
1	**NUM_ERROR_ISPS=**	xs:string 15 bytes
	Number of ISPs containing CRC errors.	xs:string 11 bytes
2	**ERROR_ISPS_THRESH=**	xs:string 18 bytes
	Threshold at which number of ISPs containing CRC errors is considered significant. For ERS, this is set to the value in the Level 0 Processor Configuration File.	xs:string 18 bytes
3	**NUM_MISSING_ISPS=**	xs:string 17 bytes
	Number of missing ISPs	xs:string 11 bytes
4	**MISSING_ISPS_THRESH=**	xs:string 20 bytes
	Number of ISPs discarded	xs:string 18 bytes
5	**NUM_DISCARDED_ISPS=**	xs:string 19 bytes
	Number of ISPs discarded by PF-HS. For ERS, always set to a default value taken from PRM_Level0SphDefaults.dat parameter file.	xs:string 11 bytes
6	**DISCARDED_ISPS_THRESH=**	xs:string 22 bytes
	Threshold at which number of ISPs discarded by PF-HS is considered significant. For ERS, this is set to the value in the Level 0	xs:string 18 bytes

#	Name/Description	Format
	Processor Configuration File.	
7	**NUM_RS_ISPS=**	xs:string 12 bytes
	Number of ISPs with Reed Solomon corrections. For ERS, always set to a default value taken from PRM_Level0SphDefaults.dat parameter file.	xs:string 11 bytes
8	**RS_THRESH=**	xs:string 10 bytes
	Number of ISPs with Reed Solomon corrections. For ERS, always set to a default value taken from PRM_Level0SphDefaults.dat parameter file.	xs:string 18 bytes
9	**Spare**	xs:string 100 bytes

Table 62: OtherProductConfidenceDataInformationType Specification

3.2.2.2.10. ASARSpecificInformationType

#	Name/Description	Format
1	**TX_RX_POLAR=**	xs:string 12 bytes
	Polarisation (used for ASAR only) HV/HV, H/HVØ, V/VHØ, H/HØØ, H/VØØ, or V/VØØ, or V/HØØ The letter(s) to the left of the ‘/’ indicates the transmitter polarisation. The letter(s) to the right of the ‘/’ indicates the receiver polarisation. ØØØØØ for non-ASAR products. Always set to V/VØØ for ERS	xs:string
2	**SWATH=**	xs:string 6 bytes
	Swath Number (used for ASAR only) codes: IS1, IS2, IS3, IS4, IS5, IS6, IS7, WSØ -- WS is used for WS mode and GM mode. For ASA_EC__OP and ASA_MS__OP, the field is set to EC0 and MC0, respectively. ØØØ for non-ASAR products. Always set to IS2 for ERS	xs:string
3	**Spare**	xs:string 41 bytes

Table 63: ASARSpecificInformationType Specification

3.2.2.2.11. ProductDSDType

#	Name/Description	Format
1	**DSD_MDS** Data Set Descriptor for the Measurement Data Set (MDS)	DSDType

Table 64: ProductDSDType Specification

3.2.2.2.12. DSDType

#	Name/Description	Format
1	**DS_NAME="**	xs:string 9 bytes
	Name describing the Data Set Left justified and filled by trailer blanks up to the length.	xs:string 28 bytes
	"	uc 1 bytes
2	**DS_TYPE=**	xs:string 8 bytes
	Type of the Data Set. Measurement (M), Reference (R) or (G) Possible values: M R G	uc
3	**FILENAME="**	xs:string 10 bytes
	Name of the Reference File. Used if DS_TYPE is set to R. It is left justified with trailer blanks Note: the file name shall be without the extension. If not used set to 62 blanks	xs:string 62 bytes
	"	uc 1 bytes
4	**DS_OFFSET=**	xs:string 10 bytes
	Offset in bytes (MPH+SPH including DSD): value<bytes>	xs:string 28 bytes
5	**DS_SIZE=**	xs:string 8 bytes
	Size in bytes of the Attached Data Set: value<bytes> Used if DS_TYPE is set to M If not used set to 00000000000000000000<bytes>	xs:string 28 bytes
6	**NUM_DSR=**	xs:string 8 bytes
	Number of Data Set Records in the Attached Data Set	int_s11d
7	**DSR_SIZE=**	xs:string 9 bytes
	Size of the Data Set Record in the Attached Data Set: value<bytes>	xs:string 18 bytes

#	Name/Description	Format
	If variable set to –0000000001<bytes> If not used set to +0000000000<bytes>	
8	**Spare**	xs:string 32 bytes

Table 65: DSDType Specification

3.2.2.2.13. SAR_IM__0P_ENVISAT_MDSType

#	Name/Description	Format
1	**Annotations**	Annotations_Type
2	**ISP**	ISP_Type

Table 66: SAR_IM__0P_ENVISAT_MDSType Specification

3.2.2.2.14. Annotations_Type

#	Name/Description	Format
1	**ISP_Sensing_Time** Level 0 Processor Annotation	mjd
2	**FEP_Annotations**	FEP_Annotations_Type

Table 67: Annotations_Type Specification

3.2.2.2.14.1.FEP_Annotations_Type

#	Name/Description	Format
1	**Blank**	xs:hexBinary 12 bytes
2	**Length_of_ISP** Length of ISP = (length of source packet) - 1	xs:hexBinary 2 bytes
3	**Blank**	xs:hexBinary 2 bytes
4	**Blank**	xs:hexBinary 2 bytes
5	**Spare**	xs:hexBinary 2 bytes

Table 68: FEP_Annotations_Type Specification

3.2.2.2.14.2.mjd

(Modified Julian Day 2000) is the decimal number of day since January 1, 2000 at 00:00 hours. It is represented by 3 long integers (4 bytes each, 12 bytes total)

#	Name/Description	Format
1	**daysElapsed** Number of days elapsed since the 1st of January 2000 at 0:0 hour. It may be negative, and is thus a signed long integer	xs:long 4 bytes
2	**secondsElapsed** Number of seconds elapsed since the beginning of that day	xs:unsignedLong 4 bytes
3	**microsecondsElapsed** Number of microseconds elapsed since the last	xs:unsignedLong 4 bytes

#	Name/Description	Format
	second	

Table 69: mjd Specification

3.2.2.2.14.3.ISP_Type

#	Name/Description	Format
1	**Data_Record_Number** Data Record Number. Starts from 1 and increments by 1 for each new MDSR in the MDS.	xs:hexBinary 4 bytes
2	**IDHT_Header** IDHT General Header	IDHT_General_Header_Type 10 bytes
3	**Auxiliary_Data_Field** Auxiliary Data and Replica/Calibration Pulses	Auxiliary_Data_Field_Type
4	**Measurement_Data** Noise and echo data, this field contains 5616 complex data samples. Each data sample is unsigned extended from the downlink format to 8-bit I and 8-bit Q.	xs:hexBinary 11232 bytes

Table 70: ISP_Type Specification

3.2.2.2.14.4.IDHT_General_Header_Type

#	Name/Description	Format
1	**Packet_counter** Packet counter	xs:hexBinary 1 bytes
2	**Subcommutation_counter** Subcommutation counter	xs:hexBinary 1 bytes
3	**IDHT_General_Header_source_packet** IDHT General Header source packet	xs:hexBinary 8 bytes

Table 71: IDHT_General_Header_Type Specification

3.2.2.2.14.5.Auxiliary_Data_Field_Type

#	Name/Description	Format
1	**Auxiliary_Data**	Auxiliary_Data_Type
2	**Cal_Rep_data**	Cal_Rep_Data_Type

Table 72: Auxiliary_Data_Field_Type Specification

3.2.2.2.14.6.Auxiliary_Data_Type

#	Name/Description	Format
1	**Format_Code** FIXED Code = AA (HEX)	xs:hexBinary 1 bytes
2	**Indication_Orbit_Ident_Code** Bit Description 0 : OGRC/OBRC Indication (1 = OBRC; 0 = OGRC) 1-4 : Orbit Ident. Code (value 0 to 15 corresponds to orbit number 1 to 16, bit 4 = LSB of code	xs:hexBinary 1 bytes
3	**ICU_Time** ICU on-board time - Continuous binary counter	xs:hexBinary 4 bytes

#	Name/Description	Format
4	**Activity_Task** Defnie the activity task within the mode of operation, accompanied by the validity flag bits and the first sample flag bits. MSB LSB 10001000 : Noise; No calibration 10011001 : No Echo; Calibration Drift (used for EM only) 10101001 : Echo; Cal. Drift 10101010 : Replica 10100000 : No Replica (because of OBRC)	xs:hexBinary 1 bytes
5	**Sample_Flags** Bit Function 0 (MSB) : Echo data (0: invalid / 1: valid) 1 : Cal. data/Replica data (0: invalid / 1: valid) 2 : Noise 3 : Cal/Repl. 4 : Echo 5 : Spare 6 : Spare 7 (LSB) : Spare	xs:hexBinary 1 bytes
6	**Image_Format_Counter** Image Format Counter updated every format. Binary counter reset at beginning of a transmission sequence. Byte 1 : Most Significative (MS) Byte 2 Byte 3 Byte 4 : Less Significative (LS) 1st format count = 0	xs:hexBinary 4 bytes
7	**Sampling_Window** Defines the time relation between the internal PRF controlpulse and the start time (i.e. first sample) of the echo sampling window. The LSB of the LS Byte shall correspond to 210.94 nsec. The timing accuracy shall be less than 60 ns Byte 1: Sampling window Byte 2: Start time	xs:hexBinary 2 bytes
8	**Pulse_Rep_Interval** Indicates the pulse repetition interval. Fixed for one operation sequence.	xs:hexBinary 2 bytes
9	**Cal_Attenuation** Specifies the attenuation setting of the calibration subsystem:	xs:hexBinary 1 bytes

#	Name/Description	Format
	Bit Description 0 : Spare 1 : 16 db attenuation step 2 : 8 db attenuation step 3 : 4 db attenuation step 4 : 2 db attenuation step 5 : 1 db attenuation step 6 : spare 7 : loop status (0=open 1=closed) Bit value = 1 (attenuation step active) Bit value = 0 (attenuation step out)	
10	**RF_Attenuation** Specifies the attenuation setting of the receive chain as it was commanded by the Macrocomand for this mode Bit Description 0 : Spare 1 : Spare 2 : 16 db attenuation step 3 : 8 db attenuation step 4 : 4 db attenuation step 5 : 2 db attenuation step 6 : 1 db attenuation step 7 : loop status (0=open 1=closed) Bit value = 1 (attenuation step active) Bit value = 0 (attenuation step out)	xs:hexBinary 1 bytes

Table 73: Auxiliary_Data_Type Specification

3.2.2.2.14.7.Cal_Rep_Data_Type

#	Name/Description	Format
1	**Pulse_sample** Calibration Pulse sample Bit 0-3 (MSB): Spare 4-9 : I 10-15 (LSB): Q	xs:hexBinary Min Occurs : 101 Max Occurs : 101 2 bytes

Table 74: Cal_Rep_Data_Type Specification

3.2.2.2.15. SAR_EWA_RecordType

#	Name/Description	Format
1	**Data_Record_Number** Data record number. starts with 1.	xs:integer 4 bytes
2	**Primary_Header**	Primary_HeaderType
3	**Secondary_Header** Secondary Header. Array of 98 unsigned bytes. This field is identical to the Secondary Header as	xs:hexBinary 98 bytes

#	Name/Description	Format
	described in ERS-2 Satellite to Ground Segment Interface Specification -ER-IS-ESA-GS-0002-December 2nd, 1993-1.1-Copyright 1993 ESA / ESTEC -	
4	**Source_Data** Source data field. Array of 4224 unsigned bytes. This field is identical to the Measurement Data Field as described in ERS-2 Satellite to Ground Segment Interface Specification -ER-IS-ESA-GS-0002-December 2nd, 1993-1.1-Copyright 1993 ESA / ESTEC -	xs:hexBinary 4224 bytes

Table 75: SAR_EWA_RecordType Specification

3.2.2.2.16. Primary_HeaderType

#	Name/Description	Format
1	**Packet_Identification** Packet Identification.	xs:integer 2 bytes
2	**Packet_Sequence_Control** Packet Sequence Control.	xs:integer 2 bytes
3	**Packet_Length** Packet Length (add 11 for the record length).	xs:integer 2 bytes

Table 76: Primary_HeaderType Specification

3.2.2.2.17. SAR_EWA_SPHType

Specific Product Header for EWAC

#	Name/Description	Format
1	**Platform_Data_Frame** This field contains 16 frames each of 256 bytes of the "Source Data Field" of the General IDHT Header Source Packet. The first 256 bytes correspond to the start of platform data, frame count of 0, and the last 256 bytes to the end of platform data, frame count of 15.	Platform_Data_FrameType Min Occurs : 16 Max Occurs : 16
2	**Ephemeris_Data** Ephemeris Data. This field is identical to the sub-commutated, 24 bytes of Ephemeris Data of the "Source Data Field" of 16 General IDHT Header Source Packets. The first 24 bytes correspond to the start of ephemeris data, ephemeris ID of 1, and the last 24 bytes to the end of ephemeris data, ephemeris ID of 16.	Ephemeris_DataType Min Occurs : 16 Max Occurs : 16

Table 77: SAR_EWA_SPHType Specification

3.2.2.2.18. Platform_Data_FrameType

#	Name/Description	Format
1	**Platform_Data**	xs:hexBinary 256 bytes

Table 78: Platform_Data_FrameType Specification

3.2.2.2.19. Ephemeris_DataType

#	Name/Description	Format
1	**Ephemeris**	xs:hexBinary 24 bytes

Table 79: Ephemeris_DataType Specification

3.3. ERS Auxiliary data (Native format)

3.3.1. Orbit files

3.3.1.1. PREC

The next figure provides a high level overview of the complex structures used to represent the information of the precise orbit file in native format:

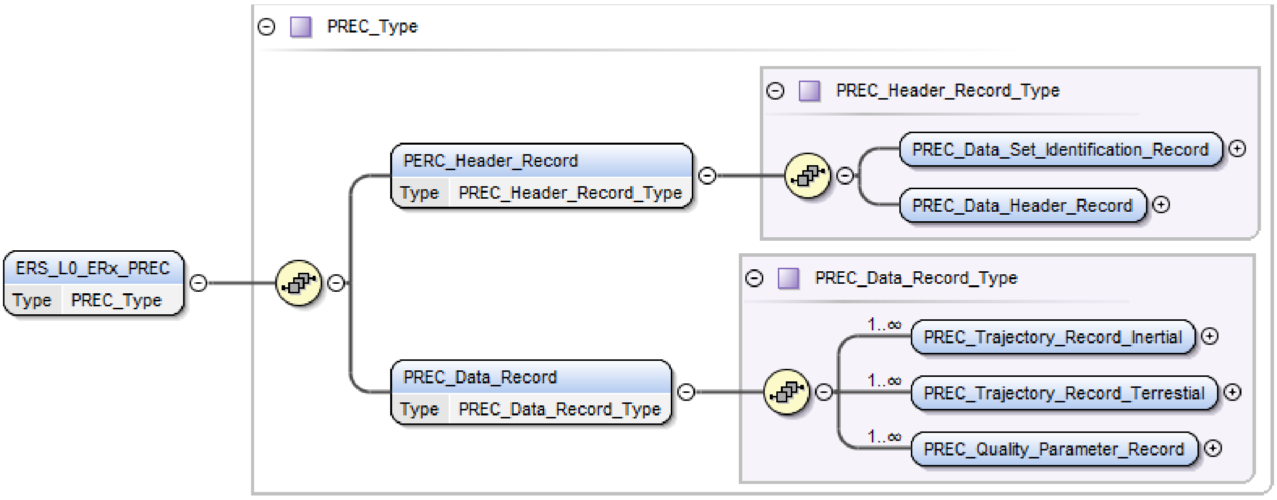

PREC_Type
PREC_Header_Record_Type
PERC_Header_Record
Type PREC_Header_Record_Type
PREC_Data_Set_Identification_Record
PREC_Data_Header_Record
ERS_L0_ERx_PREC
Type PREC_Type
PREC_Data_Record_Type
1..∞
PREC_Trajectory_Record_Inertial
1..∞
PREC_Trajectory_Record_Terrestial
1..∞
PREC_Quality_Parameter_Record
PREC_Data_Record
Type PREC_Data_Record_Type

Figure 9: DFDL schema organisation for PREC files

3.3.1.1.1. Root Element

#	Name/Description	Format
1	**ERS_L0_ERx_PREC** The precise orbit product consists of a chronologically ordered sequence (every 30s) of ASCII coded ephemerides records. Each file is preceded by a data set identification record and a data header record containing overall information about the file. The trajectory records are presented in two blocks: first in the inertial reference frame	PREC_Type

#	Name/Description	Format
	(CIS) and then in the terrestrial reference frame (CTS). At the end of the product there is a block of records containing quality information. So the product will be composed of the following records	

Table 80: ERS_L0_ERx_PREC Specification

3.3.1.1.2. Complex Types

3.3.1.1.2.1.PREC_Type

#	Name/Description	Format
1	**PERC_Header_Record**	PREC_Header_Record_Type
2	**PREC_Data_Record**	PREC_Data_Record_Type

Table 81: PREC_Type Specification

3.3.1.1.2.2.PREC_Header_Record_Type

#	Name/Description	Format
1	**PREC_Data_Set_Identification_Record**	PREC_ID_Record_Type
2	**PREC_Data_Header_Record**	PREC_Data_Header_Record_Type

Table 82: PREC_Header_Record_Type Specification

3.3.1.1.2.3.PREC_ID_Record_Type

Data Set Identification Record

#	Name/Description	Format
1	**RECKEY** DSIDP - Data Set Identification Record	xs:string 6 bytes
2	**PRODID** ERSx.ORB.PRC	xs:string 15 bytes
3	**DATTYP** POSVEL	xs:string 6 bytes
4	**SPARE** spare	xs:string 103 bytes

Table 83: PREC_ID_Record_Type Specification

3.3.1.1.2.4.PREC_Data_Header_Record_Type

PREC Data Header Record

#	Name/Description	Format
1	**RECKEY** STATE - PREC Data Header Record	xs:string 6 bytes
2	**START** Start Date of the Arc in Days since 1.1.2000 12h (days)	xs:string 6 bytes
3	**END** End Date of the Arc in Days since 1.1.2000 12h	xs:string 6 bytes

#	Name/Description	Format
	(days)	
4	**OBSTYP** Used Observation Types (3*A2) The following observation types are possible * LA - Laser * PR - Prare (range and doppler) * RA - Altimeter Ranges * XO - Altimeter Crossovers	xs:string 6 bytes
5	**OBSLEV** Used Level of Individual Observation Types (3*A2) This variable may have the following content * QL - Quick-Look or On-Site Normal Points * FR - Full-Rate * NP - Normal Points computed from Full-Rate * FD - Fast Delivery * O1 - OPR01 * MX - Mixed Types	xs:string 6 bytes
6	**MODID** Model Identifier 0 - revision 0 (PGM009): July 1991 through January 1993 (ERS-1) 1 - revision 1 (PGM035): July 1991 through March 1995 (ERS-1) 2 - revision 2 (PGM055): since July 1991 (ERS-1)/May 1995 (ERS-2)	xs:string 2 bytes
7	**RELID** Release Identifier Numerically incremented for a new release of the same product using the same model	xs:string 2 bytes
8	**RMSFIT** RMS-Fit of Orbit (mm). RMS (root mean square) of all laser residuals (observed value minus adjusted value) of the arc	xs:string 4 bytes
9	**SIGPOS** Sigma of Satellite Position (mm). Standard deviation for the estimated state vector at the start of the arc	xs:string 4 bytes
10	**SIGVEL** Sigma of Satellite Velocity (micro m/s). Standard deviation for the estimated state vector at the start of the arc	xs:string 4 bytes
11	**QUALIT** Quality Flag 0 = manoeuvre free orbit 1 = by manoeuvre degraded orbit Quality flag indicating whether the orbit is degraded by a manoeuvre (flag = 1) or not (flag = 0);	xs:string 1 bytes

#	Name/Description	Format
	flag set to 1 if manoeuvre information is available within the orbit span	
12	**TDTUTC** Time Difference TDT-UTC (s)	xs:string 5 bytes
13	**COMMNT** Comment	xs:string 78 bytes

Table 84: PREC_Data_Header_Record_Type Specification

3.3.1.1.2.5.PREC_Data_Record_Type

#	Name/Description	Format
1	**PREC_Trajectory_Record_Inertial**	PREC_Trajectory_Record_Inertial_Type Min Occurs : 1 Max Occurs : unbounded
2	**PREC_Trajectory_Record_Terrestial**	PREC_Trajectory_Record_Terrestial_Type Min Occurs : 1 Max Occurs : unbounded
3	**PREC_Quality_Parameter_Record**	PREC_Quality_Parameter_Record_Type Min Occurs : 1 Max Occurs : unbounded

Table 85: PREC_Data_Record_Type Specification

3.3.1.1.2.6.PREC_Trajectory_Record_Inertial_Type

PRC Trajectory Records (Inertial Frame)

#	Name/Description	Format
1	**RECKEY** STINER - PRC Trajectory Records (Inertial Frame)	xs:string 6 bytes
2	**SATID** Satellite ID (COSPAR No.)	xs:string 7 bytes
3	**ORBTYP** 'P' = precise	xs:string 1 bytes
4	**TTAGD** Julian Days since 1.1.2000 12h in TDT (days)	xs:string 6 bytes
5	**TTAGMS_sec** Seconds in the day	xs:string 5 bytes
6	**TTAGMS_microsec** Microseconds since 0:00 TDT (micro seconds)	xs:string 6 bytes
7	**XSAT** X-Coordinate of Satellite (CIS) (mm)	xs:string 12 bytes
8	**YSAT** Y-Coordinate of Satellite (CIS) (mm)	xs:string 12 bytes
9	**ZSAT** Z-Coordinate of Satellite (CIS) (mm)	xs:string 12 bytes
10	**XDSAT** X-Velocity of Satellite (CIS) (micro m/s)	xs:string 11 bytes
11	**YDSAT** Y-Velocity of Satellite (CIS) (micro m/s)	xs:string 11 bytes

#	Name/Description	Format
12	**ZDSAT** Z-Velocity of Satellite (CIS) (micro m/s)	xs:string 11 bytes
13	**ROLL** Roll Angle (deg)	xs:string 6 bytes
14	**PITCH** Pitch Angle (deg)	xs:string 6 bytes
15	**YAW** Yaw Angle (deg)	xs:string 6 bytes
16	**ASCARC** Flag 1 = first state vector (ascending arc) at or above the equator 0 = otherwise	xs:string 2 bytes
17	**CHECK** Checksum of Cols 21-120. The checksum is computed per record by using the following rules: sum of all single digits in columns 21 through 120 (sign,dot not being considered).	xs:string 3 bytes
18	**QUALI** Quality Flag 0 = good quality 1 = by manoeuvre degraded state	xs:string 1 bytes
19	**RADCOR** Radial Orbit Correction (cm). This correction will improve the orbit quality in radial direction and when being applied it has to be subtracted from the satellite height. The value is set to 9999, 9998 or 9997 in the following cases: * there are gaps in the altimeter data so that no correction can be computed (will affect one half-revolution) (value=9999) * the position is over land (value=9998). This flagging is not performed for the first and last state over land in order to support the interpolation of values. * the correction exceeds a threshold of 3 times the crossover rms, i.e. a threshold of 60 cm (value=9997)	xs:string 4 bytes
20	**SPARE** spare	xs:string 2 bytes

Table 86: PREC_Trajectory_Record_Inertial_Type Specification

3.3.1.1.2.7.PREC_Trajectory_Record_Terrestial_Type

PREC Trajectory Records (Terrestrial Frame)

#	Name/Description	Format
1	**RECKEY** STTERR - PRC Trajectory Records (Terrestrial Frame)	xs:string 6 bytes
2	**SATID** Satellite ID (COSPAR No.)	xs:string 7 bytes
3	**OBSTYP** 'P' = precise	xs:string 1 bytes
4	**TTAGD** Julian Days since 1.1.2000 12h in TDT (days)	xs:string 6 bytes
5	**TTAGMS** Microseconds since 0:00 TDT (micro seconds)	xs:string 11 bytes
6	**XSAT** X-Coordinate of Satellite (CIS) (mm)	xs:string 12 bytes
7	**YSAT** Y-Coordinate of Satellite (CIS) (mm)	xs:string 12 bytes
8	**ZSAT** Z-Coordinate of Satellite (CIS) (mm)	xs:string 12 bytes
9	**XDSAT** X-Velocity of Satellite (CIS) (micro m/s)	xs:string 11 bytes
10	**YDSAT** Y-Velocity of Satellite (CIS) (micro m/s)	xs:string 11 bytes
11	**ZDSAT** Z-Velocity of Satellite (CIS) (micro m/s)	xs:string 11 bytes
12	**ROLL** Roll Angle (deg)	xs:string 6 bytes
13	**PITCH** Pitch Angle (deg)	xs:string 6 bytes
14	**YAW** Yaw Angle (deg)	xs:string 6 bytes
15	**ASCARC** Flag 1 = first state vector (ascending arc) at or above the equator 0 = otherwise	xs:string 2 bytes
16	**CHECK** Checksum of Cols 21-120 The checksum is computed per record by using the following rules: sum of all single digits in columns 21 through 120 (sign,dot not being considered).	xs:string 3 bytes
17	**QUALI** Quality Flag 0 = good quality 1 = by manoeuvre degraded state Quality flag indicating whether the state vector is degraded by a manoeuvre (flag = 1) or not (flag = 0)	xs:string 1 bytes
18	**RADCOR**	xs:string

#	Name/Description	Format
	Radial Orbit Correction (cm) This correction will improve the orbit quality in radial direction and when being applied it has to be subtracted from the satellite height. The value is set to 9999, 9998 or 9997 in the following cases: * there are gaps in the altimeter data so that no correction can be computed (will affect one half-revolution) (value=9999) * the position is over land (value=9998). This flagging is not performed for the first and last state over land in order to support the interpolation of values. * the correction exceeds a threshold of 3 times the crossover rms, i.e. a threshold of 60 cm (value=9997)	4 bytes
19	**SPARE** spare	xs:string 2 bytes

Table 87: PREC_Trajectory_Record_Terrestial_Type Specification

3.3.1.1.2.8.PREC_Quality_Parameter_Record_Type

Quality Parameter Records

#	Name/Description	Format
1	**RECKEY** QUALCO - Quality Parameter Records	xs:string 6 bytes
2	**Space** Separator	xs:string 1 bytes
3	**QPNAME** Quality Parameter Name: * Sigma_Unit_Weight: standard deviation of the unit weight; computation by the square root of the sum of the weighted squared observation residuals divided by the degrees of freedom * Sta_Dev_Obs_Laser: standard deviation of laser residuals of the orbit * Mean_Obs_Laser: mean of laser residuals of the orbit * Sta_Dev_Obs_Crossover: standard deviation of altimeter crossover residuals of the orbit * Mean_Obs_Crossover: mean of altimeter crossover residuals of the orbit	xs:string 24 bytes

#	Name/Description	Format
	* Sta_Dev_Obs_Prare_Ran standard deviation of Prare range residuals of the orbit * Mean_Obs_Prare_Ran mean of Prare range residuals of the orbit * Sta_Dev_Obs_Prare_Dop standard deviation of Prare doppler residuals of the orbit * Mean_Obs_Prare_Dop mean of Prare doppler residuals of the orbit * RMS_Orb_Along_Track_XXXX (XXXX: Begi, Mid or End) rms of differences in along-track direction of the actual arc and an arc overlapping the begin, mid or end of the actual orbit * RMS_Orb_Cross_Track_XXXX (XXXX: Begi, Mid or End) rms of differences in cross-track direction of the actual arc and an arc overlapping the begin, mid or end of the actual orbit * RMS_Orb_Radial_XXXX (XXXX: Begin, Mid or End) rms of differences in radial direction of the actual arc and an arc overlapping the begin, mid or end of the actual orbit * Sta_Dev_Cro_Corr_MProd standard deviation of altimeter crossover residuals of the orbit after applying the radial orbit correction for the "monthly" product * Mean_Cro_Corr_MProd mean of altimeter crossover residuals of the orbit after applying the radial orbit correction	
4	**Space** Separator	xs:string 1 bytes
5	**EQSIGN** =	xs:string 3 bytes
6	**QPVALUE** Quality Parameter Value	xs:string 10 bytes
7	**QPUNIT** Units of QPVALUE	xs:string 24 bytes

#	Name/Description	Format
8	**QPREFVAL** Reference Value	xs:string 10 bytes
9	**SPARE** spare	xs:string 51 bytes

Table 88: PREC_Quality_Parameter_Record_Type Specification

3.3.1.2. ORPM

The next figure provides a high level overview of the complex structures used to represent the information of the Predicted Orbit in native format:

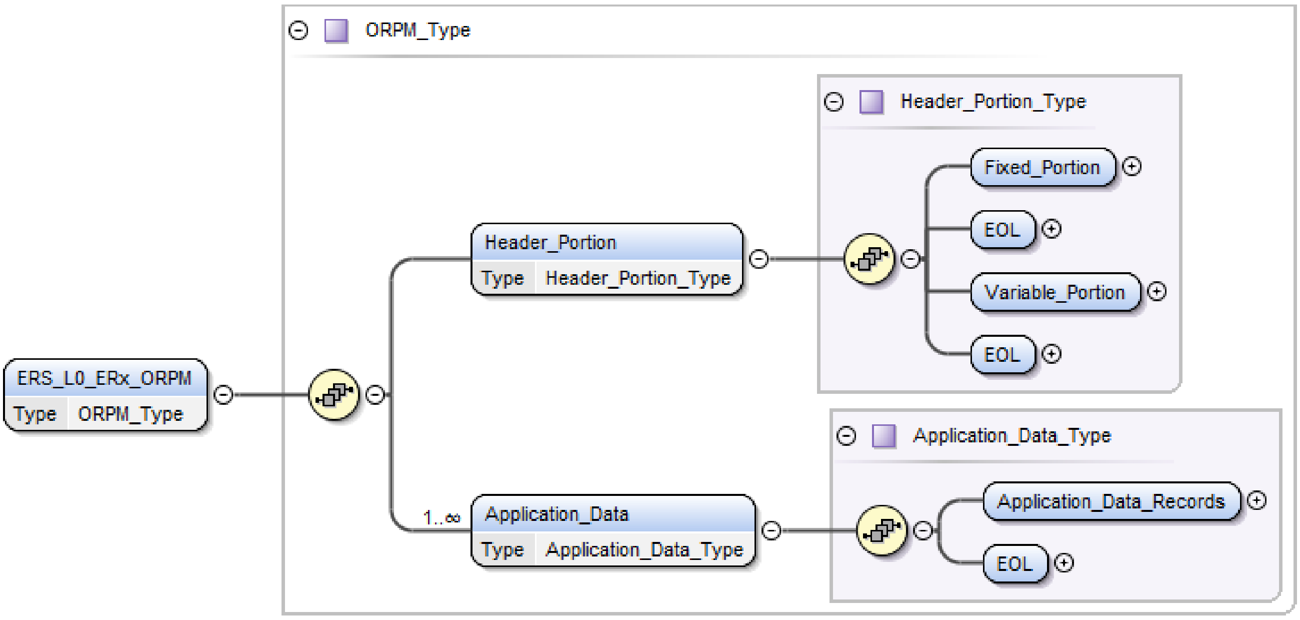


Figure 10: DFDL schema organisation for ORPM files

3.3.1.2.1. Root Element

#	Name/Description	Format
1	**ERS_L0_ERx_ORPM** The Predicted Orbit information is extracted from the file Predicted MMCC Orbit, generated at MMCC after the last of consecutive passes over Kiruna, containing one predicted state vector per orbit, at the ascending node, for the coming 6 days.	ORPM_Type

Table 89: ERS_L0_ERx_ORPM Specification

3.3.1.2.2. Complex Types

3.3.1.2.2.1.ORPM_Type

#	Name/Description	Format
1	**Header_Portion**	Header_Portion_Type
2	**Application_Data**	Application_Data_Type Min Occurs : 1 Max Occurs : unbounded

Table 90: ORPM_Type Specification

3.3.1.2.2.2.Header_Portion_Type

#	Name/Description	Format
1	**Fixed_Portion**	Fixed_Portion_Type
2	**EOL**	xs:string 1 bytes

#	Name/Description	Format
3	**Variable_Portion**	Variable_Portion_Type
4	**EOL**	xs:string 1 bytes

Table 91: Header_Portion_Type Specification

3.3.1.2.2.3.Fixed_Portion_Type

#	Name/Description	Format
1	**File_identifier**	xs:string 5 bytes
2	**Generation_Date** Format: YYMMDD	xs:string 6 bytes
3	**Originator**	xs:string 2 bytes
4	**Destination**	xs:string 2 bytes
5	**Cyclic_Counter**	xs:string 4 bytes
6	**Separator**	xs:string 1 bytes
7	**Satellite_ID**	xs:string 2 bytes
8	**Generation_Time**	xs:string 8 bytes

Table 92: Fixed_Portion_Type Specification

3.3.1.2.2.4.Variable_Portion_Type

#	Name/Description	Format
1	**Filler_1** Filler (always zero) (6 ASCII digits)	xs:string 6 bytes
2	**Filler_2** Filler (always zero) (10 ASCII digits)	xs:string 10 bytes
3	**Start_Time_days** Start time in integer days (MJD) (6 ASCII digits)	xs:string 6 bytes
4	**Start_Time_msec** Start time in msec since midnight UTC(10 ASCII digits)	xs:string 10 bytes
5	**End_Time_days** End time in integer days (MJD) (6 ASCII digits)	xs:string 6 bytes
6	**End_Time_msec** End time in msec since midnight UTC (10 ASCII digits)	xs:string 10 bytes
7	**Last_Update** Last update (14 ASCII characters)	xs:string 33 bytes

Table 93: Variable_Portion_Type Specification

3.3.1.2.2.5.Application_Data_Type

#	Name/Description	Format
1	**Application_Data_Records**	Application_Data_Records_Type

#	Name/Description	Format
2	**EOL**	xs:string 1 bytes

Table 94: Application_Data_Type Specification

3.3.1.2.2.6.Application_Data_Records_Type

#	Name/Description	Format
1	**Node_Crossing_Time_Days** Node crossing time in integer days (MJD)	xs:string 6 bytes
2	**Node_Crossing_Time_Msec** Node crossing time in msec since midnight UTC	xs:string 10 bytes
3	**Orbit_Nb** Orbit number	xs:string 7 bytes
4	**Position_Vectors** Position Vectors (x, y, z) Km in F15.6 FORTRAN format. The reference frame for the state vectors is the Conventional Terrestrial System (CTS), defined as: *) X axis in the equatorial plane from the Earth centre to the zero longitude meridian (Greenwich); *) Y axis completing the left handed reference frame. *) Z axis from the Earth centre to the pole, coinciding with the Earth rotation axis as defined in the latest annual report of the International Earth Rotation Service;	xs:string Min Occurs : 3 Max Occurs : 3 15 bytes
5	**Velocity_Vectors** Velocity Vectors (dx/dt, dy/dt, dz/dt) Km/s in F15.9 FORTRAN format. The reference frame for the state vectors is the Conventional Terrestrial System (CTS), defined as: *) Z axis from the Earth centre to the pole, coinciding with the Earth rotation axis as defined in the latest annual report of the International Earth Rotation Service; *) X axis in the equatorial plane from the Earth centre to the zero longitude meridian (Greenwich); *) Y axis completing the left handed reference frame.	xs:string Min Occurs : 3 Max Occurs : 3 15 bytes
6	**TAI_UTC_Delta** TAI-UTC Delta The TAI-UTC Delta is the value of the difference between the International Atomic Time (TAI) and the Universal Time Coordinated (UTC) in seconds. Normally this value is zero. Only when the two clocks are not in synchronism, this value can be different from zero. In this case it	xs:string 2 bytes

#	Name/Description	Format
	is foreseen that it will normally be +1 second and exceptionally +2 seconds.	

Table 95: Application_Data_Records_Type Specification

3.3.1.3. ORRM

The next figure provides a high level overview of the complex structures used to represent the information of the Restituted Orbit file in native format format:

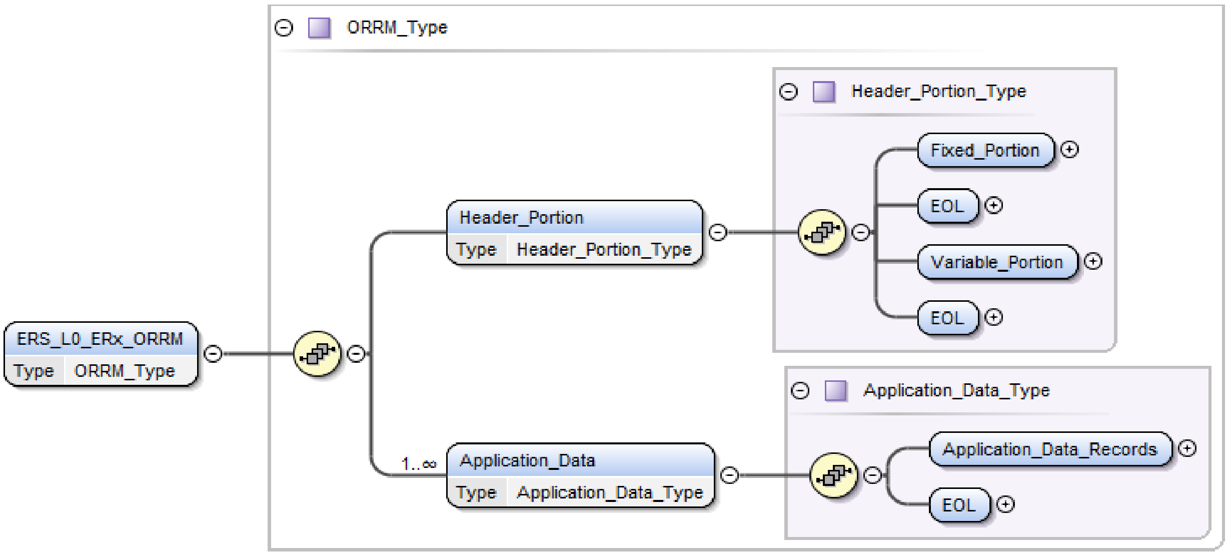


Figure 11: DFDL schema organisation for ORRM files

3.3.1.3.1. Root Element

#	Name/Description	Format
1	**ERS_L0_ERx_ORRM** The Restituted_Orbit is extracted from a file generated at MMCC called Restituted_MMCC_Orbit, containing 1 state vector per minute, for one day with a delay of 3 days. This Restituded_MMCC_Oribt file is generated after the last of consecutive passes over Kiruna and is optionally available for transmission to the PAFs.	ORRM_Type

Table 96: ERS_L0_ERx_ORRM Specification

3.3.1.3.2. Complex Types

3.3.1.3.2.1.ORRM_Type

#	Name/Description	Format
1	**Header_Portion**	Header_Portion_Type

#	Name/Description	Format
2	**Application_Data**	Application_Data_Type Min Occurs : 1 Max Occurs : unbounded

Table 97: ORRM_Type Specification

3.3.1.3.2.2.Header_Portion_Type

#	Name/Description	Format
1	**Fixed_Portion**	Fixed_Portion_Type
2	**EOL**	xs:string 1 bytes
3	**Variable_Portion**	Variable_Portion_Type
4	**EOL**	xs:string 1 bytes

Table 98: Header_Portion_Type Specification

3.3.1.3.2.3.Fixed_Portion_Type

#	Name/Description	Format
1	**File_identifier**	xs:string 5 bytes
2	**Generation_Date** Format: YYMMDD	xs:string 6 bytes
3	**Originator**	xs:string 2 bytes
4	**Destination**	xs:string 2 bytes
5	**Cyclic_Counter**	xs:string 4 bytes
6	**Separator**	xs:string 1 bytes
7	**Satellite_ID**	xs:string 2 bytes
8	**Generation_Time** Format: hh:mm:ss	xs:string 8 bytes

Table 99: Fixed_Portion_Type Specification

3.3.1.3.2.4.Variable_Portion_Type

#	Name/Description	Format
1	**Filler_1** Filler (always zero) (6 ASCII digits)	xs:string 6 bytes
2	**Filler_2** Filler (always zero) (10 ASCII digits)	xs:string 10 bytes
3	**Start_Time_days** Start time in integer days (MJD) (6 ASCII digits)	xs:string 6 bytes
4	**Start_Time_msec** Start time in msec since midnight UTC(10 ASCII digits)	xs:string 10 bytes
5	**End_Time_days** End time in integer days (MJD) (6 ASCII digits)	xs:string 6 bytes

#	Name/Description	Format
6	**End_Time_msec** End time in msec since midnight UTC (10 ASCII digits)	xs:string 10 bytes
7	**Last_Update** Last update (14 ASCII characters)	xs:string 33 bytes

Table 100: Variable_Portion_Type Specification

3.3.1.3.2.5.Application_Data_Type

#	Name/Description	Format
1	**Application_Data_Records**	Application_Data_Records_Type
2	**EOL**	xs:string 1 bytes

Table 101: Application_Data_Type Specification

3.3.1.3.2.6.Application_Data_Records_Type

#	Name/Description	Format
1	**Time_Days** Time in integer days (MJD)	xs:string 6 bytes
2	**Time_Msec** Time in msec since midnight UTC	xs:string 10 bytes
3	**Orbit_Nb** Orbit number	xs:string 7 bytes
4	**Position_Vectors** Position Vectors (x, y, z) Km in F15.6 FORTRAN format. The reference frame for the state vectors is the Conventional Terrestrial System (CTS), defined as: *) X axis in the equatorial plane from the Earth centre to the zero longitude meridian (Greenwich); *) Y axis completing the left handed reference frame. *) Z axis from the Earth centre to the pole, coinciding with the Earth rotation axis as defined in the latest annual report of the International Earth Rotation Service;	xs:string Min Occurs : 3 Max Occurs : 3 15 bytes
5	**Velocity_Vectors** Velocity Vectors (dx/dt, dy/dt, dz/dt) Km/s in F15.9 FORTRAN format. The reference frame for the state vectors is the Conventional Terrestrial System (CTS), defined as: *) Z axis from the Earth centre to the pole, coinciding with the Earth rotation axis as defined in the latest annual report of the International Earth Rotation Service; *) X axis in the equatorial plane from the Earth	xs:string Min Occurs : 3 Max Occurs : 3 15 bytes

#	Name/Description	Format
	centre to the zero longitude meridian (Greenwich); *) Y axis completing the left handed reference frame.	
6	**TAI_UTC_Change** TAI-UTC Change The TAI-UTC Change is normally zero, but it assumes the value (in seconds) of the difference between the International Atomic Time (TAI) and the Universal Time Coordinated (UTC), when a leap second is detected. Therefore this flag will be set, if neccessary, only at the midnights of 31 December and 30 June, when a leap second might be detected. When set, it is expected to assume a positive value of 1 second, normally, and of 2 seconds, excepcionally.	xs:string 2 bytes

Table 102: Application_Data_Records_Type Specification

3.3.2. Time Correlation file

3.3.2.1. PATC

The next figure provides a high level overview of the complex structures used to represent the information of the Time Correlation file in native format:

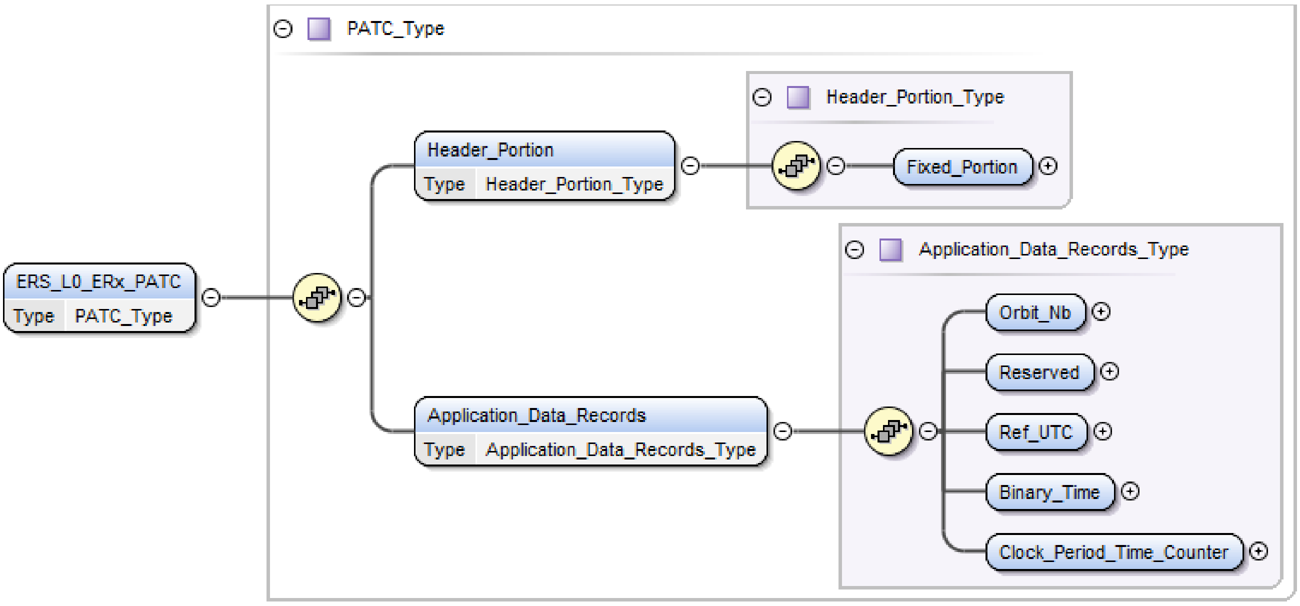


Figure 12: DFDL schema organisation for PATC files

3.3.2.1.1. Root Element

#	Name/Description	Format
1	**ERS_L0_ERx_PATC** EECF transmits per orbit (when available) to EGSs except Kiruna one file containing one Time	PATC_Type

#	Name/Description	Format
	Correlation element, which permits to correlate the satellite's on board time with the UTC one. The EGSs will use only the most recent of the time correlation elements provided by MMCC for inclusion in the detailed processing schedules. It should be noted that a time correlation element is not available for each of the orbits.	

Table 103: ERS_L0_ERx_PATC Specification

3.3.2.1.2. Complex Types

3.3.2.1.2.1.PATC_Type

#	Name/Description	Format
1	**Header_Portion**	Header_Portion_Type
2	**Application_Data_Records**	Application_Data_Records_Type

Table 104: PATC_Type Specification

3.3.2.1.2.2.Header_Portion_Type

#	Name/Description	Format
1	**Fixed_Portion**	Fixed_Portion_Type

Table 105: Header_Portion_Type Specification

3.3.2.1.2.3.Fixed_Portion_Type

#	Name/Description	Format
1	**File_identifier**	xs:string 5 bytes
2	**Generation_Date**	xs:string 6 bytes
3	**Originator**	xs:string 2 bytes
4	**Destination**	xs:string 2 bytes
5	**Cyclic_Counter**	xs:string 4 bytes
6	**Separator**	xs:string 1 bytes
7	**Satellite_ID**	xs:string 2 bytes
8	**Generation_Time**	xs:string 8 bytes

Table 106: Fixed_Portion_Type Specification

3.3.2.1.2.4.Application_Data_Records_Type

#	Name/Description	Format
1	**Orbit_Nb** Orbit number	xs:string 5 bytes
2	**Reserved** Reserved	xs:string 3 bytes

#	Name/Description	Format
3	**UTC_Ndays** Reference UTC Time (microseconds)	xs:int 4bytes
4	**UTC_Time_of_day** Reference UTC Time (microseconds)	xs:int 4bytes
5	**Binary_Time** Satellite Time Binary Counter Value	xs:unsignedInt 4bytes
6	**Clock_Period_Time_Counter** Clock Period of Satellite Time Counter (nano seconds)	xs:unsignedInt 4bytes

Table 107: Application_Data_Records_Type Specification